

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

**USING XML/HTTP TO STORE, SERVE AND ANNOTATE
TACTICAL SCENARIOS FOR X3D OPERATIONAL
VISUALIZATION AND ANTI-TERRORIST TRAINING**

by

Khaled Mnif

March 2003

Thesis Advisor:
Thesis Co-advisor:

Don Brutzman
Curtis L. Blais

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2003	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Using XML/HTTP to Store, Serve and Annotate Tactical Scenarios for X3D Operational Visualization and Anti-Terrorist Training			5. FUNDING NUMBERS	
6. AUTHOR Khaled Mnif				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT <p>Adopting Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) are key steps to accommodate the evolution of Internet technologies. While HTTP is already a proven standard communication protocol responsible for the rapid expansion of the World Wide Web, XML provides general mechanisms for determining validatable documents and addresses several deficiencies of HTML regarding diverse document structure and content. XML and HTTP together provide many of the essential capabilities associated with database engines.</p> <p>The Modeling, Virtual Environments and Simulation (MOVES) Institute of the Naval Postgraduate School (NPS) is continuing to build a database of 3D tactical scenarios and using X3D and VRML tools. The configuration parameters and statistical results of these scenarios are XML documents. For a better understanding and usability of these results by the end users, a Webbased application stores and manipulates these XML document.</p> <p>This thesis develops a server-side application that can store, serve, and annotate tactical scenarios for X3D operational visualization and anti-terrorist training by using XML and HTTP technologies. The experimental demonstration for this work is the prototypical Anti-Terrorism/Force Protection (AT/FP) simulation model developed by Lieutenant James W. Harney, USN, using Extensible 3D Graphics (X3D)/ Virtual Reality Modeling Language (VRML) models.</p>				
13. SUBJECT TERMS Extensible Markup Language XML, Hypertext Transfer Protocol HTTP, Extensible 3D Graphics X3D, Extensible Stylesheet Language Transformations XSLT, Scalable Vector Graphics SVG, Force Protection, Anti-Terrorism, SAVAGE			15. NUMBER OF PAGES 136	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**USING XML/HTTP TO STORE, EDIT, SERVE AND ANNOTATE
TACTICAL SCENARIOS FOR X3D OPERATIONAL VISUALIZATION AND
ANTI-TERRORIST TRAINING**

Khaled Mnif
Captain, Tunisian Army
B.S., FSEG Sfax, 1985

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
March 2003**

Author:

Khaled Mnif

Approved by:

Don Brutzman, Thesis Advisor

Curtis L. Blais, Thesis Co-Advisor

Thomas Otani, Academic Associate

Peter Denning, Chairman
Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Adopting Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) are key steps to accommodate the evolution of Internet technologies. While HTTP is already a proven standard communication protocol responsible for the rapid expansion of the World Wide Web, XML provides general mechanisms for determining validatable documents and addresses several deficiencies of HTML regarding diverse document structure and content. XML and HTTP together provide many of the essential capabilities associated with database engines

XML enables the creation of web documents that preserve data structure and simultaneously provide human-readable and machine-readable information to facilitate web automation. Today XML can guarantee platform-independent inter-application data interchange. XML is becoming an enabling technology on the Internet, transforming the Web from a static medium to a powerful infrastructure for collaborative and interoperable applications.

The Modeling, Virtual Environments and Simulation (MOVES) Institute of the Naval Postgraduate School (NPS) is continuing to build a database of 3D tactical scenarios and using X3D and VRML tools. The configuration parameters and statistical results of these scenarios are XML documents. For a better understanding and usability of these results by the end users, a Webbased application stores and manipulates these XML document.

This thesis develops a server-side application that can store, serve, and annotate tactical scenarios for X3D operational visualization and anti-terrorist training by using XML and HTTP technologies. The experimental demonstration for this work is the prototypical Anti-Terrorism/Force Protection (AT/FP) simulation model developed by Lieutenant James W. Harney, USN, using Extensible 3D Graphics (X3D)/ Virtual Reality Modeling Language (VRML) models.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION.....	1
A. PROBLEM STATEMENT	1
B. MOTIVATION.....	1
C. OBJECTIVES AND APPROACH.....	2
D. ORGANIZATION OF THESIS	2
II. RELATED WORK.....	3
A. INTRODUCTION.....	3
B. EXTENSIBLE MARKUP LANGUAGE (XML)	3
C. EXTENSIBLE 3D (X3D) GRAPHICS.....	5
D. MAJ. CLIF WILLIAMS' THESIS: HTTP/XML SERVING	7
E. MAJ. SHANE NICKLAUS' THESIS: XML OPORDER VISUALIZATION	8
F. LT. JAMES HARNEY'S THESIS: AGENT-BASED ANTI-TERRORISM/FORCE PROTECTION MODEL	8
G. HTTP AND APACHE WEB SERVER SERVING XML.....	9
H. SUMMARY	10
III. EXAMPLE MILITARY TACTICAL DATABASES	11
A. INTRODUCTION.....	11
B. PARAMETERS OF INTEREST INCLUDING ORIGIN AND METADATA	11
1. Compare Static To Dynamic Database Origin.....	11
C. NATO BATTLESPACE GENERIC HUB FOR (BGH)	12
D. TACTICAL DATA SOURCES:	13
1. FAS -USNI.....	13
- Capabilities, Limitations, and Availability	
2. Roland & Associates (R & A)	15
3. DIS Protocol Enumeration.....	16
4. Military Symbolology Library.....	18
E. CONVERTING DATASETS AND DATABASES TO XML	20
1. XML Versus Database.....	20
2. XML From Databases And Vice Versa	20
3. XML-Enabled Databases	25
4. XML Servers	25
5. Native XML Databases.....	27
6. XML Query Engines.....	31
F. DESIGN GOALS AND LESSONS LEARNED	32
G. SUMMARY	32
IV. TOOLS, ARCHITECTURE AND WORK FLOW.....	33
A. INTRODUCTION.....	33
B. DEVELOPMENT TOOLS	33
1. XML SPY.....	33
2. Saxon: An XSLT Processor.....	36
3. Xerces: XML Parser	38

4. HTML-Kit	40
5. Java Servlets	41
C. ARCHITECTURE OF SYSTEM COMPONENTS	43
D. SUMMARY	43
V. TACTICAL SCENARIO DESCRIPTION.....	45
A. INTRODUCTION.....	45
B. SCENARIO COMPONENTS	45
1. Scenario's Entities	45
2. Location	47
C. REQUIREMENTS FOR SCENARIO CREATION.....	48
D. REQUIREMENTS FOR STORING SCENARIO RESULTS.....	50
E. SUMMARY.....	50
VI. XML SCHEMA AND XSLT STYLESHEETS FOR ANTI-TERRORIST SCENARIO REPRESENTATION.....	51
A. INTRODUCTION.....	51
B. XML SCHEMA FOR ANTI-TERRORIST SCENARIO.....	51
1. Friendly And Hostile Forces, Location, And Measures Of Effectiveness	51
C. XSLT STYLESHEETS FOR ANTI-TERRORIST SCENARIO	53
1. XSLT Stylesheet for Entities Properties and Tactical Data	53
2. XSLT Stylesheet for Measure Of Effectiveness	53
D. SUMMARY	53
VII. EXPERIMENTAL AND SIMULATION RESULTS.....	55
A. INTRODUCTION.....	55
B. AT/FP SCENARIO INSTANCE.....	55
C. SCENARIO SCREEN CAPTURES EXPLAINED	55
1. User Interface	55
2. Simulation Results	56
a. Scenario Properties and Tactical Data	56
b. Scenario Statistic Output	57
D. SUMMARY	58
VIII. CONCLUSIONS AND RECOMMENDATIONS	59
A. CONCLUSIONS	59
B. RECOMMENDATIONS FOR FUTURE WORK.....	60
APPENDIX A. AT/FP XML SCHEMA.....	61
APPENDIX B. XSLT STYLESHEET FOR ENTITY PROPERTIES AND TACTICAL DATA.....	89
APPENDIX C. XSLT STYLESHEET FOR MESURE OF EFFECTIVENESS.....	103
APPENDIX D. AT/FP SCENARIO INSTANCE.....	113
LIST OF REFERENCES	121
INITIAL DISTRIBUTION LIST	123

LIST OF FIGURES

Figure II.1.	Screen Capture of the X3D Edit Tool.....	6
Figure III.1.	Generic Hub and Its Relationship to Sub functional Areas.....	13
Figure III.2.	Depiction of DDG-51, Arleigh Burke class destroyer.	14
Figure III.3.	Screen Snapshot of the complete Arleigh Burke Class X3D Model.....	15
Figure III.4.	The relationship between the component documents comprising the set of IEEE DIS documents.....	17
Figure III.5	Common warfighting symbology documents.	19
Figure III.6	Methods of converting database data to XML	31
Figure IV.1.	Screen Capture of the three main areas of XSLT Designer	34
Figure IV.2.	Screen Capture of the menu and the main window of XML Spy IDE.....	36
Figure IV.3.	Saxon architecture	37
Figure IV.4.	Screen Capture of HTML-KIT tool in action.....	41
Figure IV.5.	Java Servlet Engine and Web Server Relationship	42
Figure IV.6.	Architecture of system components	43
Figure V.1.	Screen Snapshot of the RHIB used for ATFP Scenario	46
Figure V.2.	Screen Snapshot view of the Terrorist Boat used to attack USS COLE	47
Figure V.3.	Screen capture of Aden Harbor, Yemen X3D scene	48
Figure V.4.	Example of the terrorist boat attack profile setup.....	49
Figure VI.1.	Schema design view of ATFP	52
Figure VII.1.	Screen Capture of the user interface	56
Figure VII.2.	Screen Capture displaying the properties and tactical data of an XML instance for AT/FP sceario run	57
Figure VII.3.	Screen Capture displaying measure of effectiveness (MOE) of the AT/FP sceario runs	58

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table III.1.	List of products enabling the conversion between XML and Databases.....	22
Table III.2.	List of XML-Enabled Databases	25
Table III.3.	List of XML Server products.....	26
Table III.4.	List of XML Databases.....	29

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to dedicate all my work and efforts at the Naval Postgraduate School in Monterey in memory of my unforgettable and great mother Khadija.

Second, many thanks to:

- Don Brutzman and Curtis Blais from the Naval Postgraduate School (NPS) for their guidance, support and professionalism in allowing me to take an idea and turn it into a learning opportunity.
- The entire SAVAGE team for their camaraderie and sharing of ideas and exemplar work.
- My wife Nissaf, and my baby Amin for their encouragement, tolerance, and perseverance during my staying in Monterey and thesis completion.

I. INTRODUCTION

A. PROBLEM STATEMENT

The Modeling, Virtual Environments and Simulation (MOVES) Institute of the Naval Postgraduate School (NPS) possesses a growing database of 3D models and scenarios. In particular, a Tree-Dimensional (3D) Anti-Terrorism/Force Protection (AT/FP) scenario was developed in parallel with the present research. [Harney 2003]. This research demonstrates the functionality of Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) with other Web-based tools in order to serve, visualize, store and annotate 3D Scenarios. XML and HTTP together provide many of the essential capabilities associated with database engines

The basic web-based tools needed for performing the different tasks of this research are:

- Extensible Markup language (XML) for creating instances of XML documents
- XML Schema used to define valid documents and records
- Extensible Style-sheet Language Transformations (XSLT) for integration and conversion of scenario data
- Extensible 3D Graphics (X3D) to visualize scenarios
- HTTP Server for database access; and
- HTML-kit used to design and prototype HTML forms for results annotation.

B. MOTIVATION

To be successful in preventing and avoiding acts of terror against military targets, military units must be able to quickly get the right information necessary in order to make a decision and react at the right time. Hence, it is desirable to up the capability and with widespread support of the XML technology standard for describing and sharing structured information, adopting the HTTP based on the client/server principle could be the best combination to store, edit, serve and annotate scenarios in order to make optimal decisions in short time. Important results are thus possible:

- Semantically correct and comprehensible information
- Validatable messages for use by humans and systems

- Easily stored and retrievable. When carefully constructed, it is compatible with diverse command and control systems.

C. OBJECTIVES AND APPROACH

This thesis applies Web-based technologies to store, edit, serve, and annotate tactical scenarios. The particular objectives of the thesis are augmenting the capabilities of the AT/FP [Harney 2003] and supporting X3D applications for operational visualization and anti-terrorist training. To accomplish these objectives, several tasks have to be performed. First, an XML Schema representing the data structure of the AT/FP scenario must be created. This structure has to describe all the entities (e.g. ships, RHIB, terrorist, neutral, shipping; etc.), location (e.g. name, position...), and statistics output. Next, a representation of NATO's Generic Hub (GH) using XML Schema elements is performed. Then, the common elements and possibilities between our AT/FP scenario and NATO's GH are identified. Finally, a web-based application on the server side and an end-user interface on the client are developed, using web-based languages and software tools.

D. ORGANIZATION OF THESIS

Chapter II summarizes related work from previous theses that apply XML technologies with X3D and VRML to build, distribute, and visualize three-dimensional (3D) simulation. The chapter also introduces the basic tools used to perform this research. Chapter III presents examples of databases containing military tactical parameters, compares regular databases to XML, and identifies different products enabling the conversion of XML from databases and vice versa. Chapter IV defines the principal tools used to perform tasks needed for this research. It also describes the system architecture of the application. Chapter V describes the components of the specific Anti-Terrorist/Force Protection (AT/FP) scenario simulating the attack on the USS COLE. Chapter VI defines the XML Schema structuring the ATFP XML tactical results document, and provides examples of XSLT stylesheets transforming instances of XML document. Chapter VII contains an example XML document treated as an instance of our XML database, and provides simulation results. Chapter VIII provides conclusions for this research and recommendations for future work.

II. RELATED WORK

A. INTRODUCTION

This chapter first summarizes related work applying XML technologies with X3D and VRML to build, distribute, and visualize three-dimensional (3D) simulation. It then introduces the basic languages such as XML, HTTP and X3D. Numerous examples of related work synopsized in order to provide essential background and resources for additional information.

B. EXTENSIBLE MARKUP LANGUAGE (XML)

The Standard Generalized Markup Language (SGML) was an early attempt in 1980 to combine a universally interchangeable data format with rich information storage capabilities. SGML was designed to be a standard way of marking up data for any purpose. HyperText Markup Language (HTML) uses many of SGML's concepts to provide a universal markup language for the display of information and the linking of different pieces of information. SGML is a complicated language, not well suited for data interchange over the web [HUNTER 01]. HTML has the limit action of providing a fixed set of tags that do not allow description of the content they encompass. XML is a subset of SGML and became a World Wide Web Consortium (W3C) Recommendation in February 1998. It was created to reduce the complexity of SGML and address the limitations of HTML's Tags. The reference <http://www.w3.org/XML/1999/XML-in-10-points> "XML in Ten Points" is a summary that can help beginners to understand the basic concepts of XML. A paraphrased description follows, adapted from that reference.

1. XML is for Structuring Data

XML is a set of rules for designing text formats in order to ensure that there is no ambiguity in the data structure.

2. XML Looks a bit Like HTML

Both HTML and XML use tags (words bracketed by '<' and '>') and attributes (of the form name="value"). The difference is that with HTML, each tag and attribute has a specific meaning, and often the content between them will be displayed in a browser, but

XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it.

3. XML is Text, But is Not Meant to be Read

Similar to HTML, XML files are text files that people should not have to read, but if the need arises, any may use their favorite text editor to read them.

4. XML is Verbose by Design

The verbosity of XML resides into its large text formatted files compared to binary formatted files. However, this disadvantage is compensated at a different level:

- Disk space is less expensive than it used to be
- The existence of effective and fast compression programs (zip, gzip)
- The efficiency of communication protocols (such as modem protocols and HTTP/1.1,) to compress data on the fly, saving bandwidth is as effectively as a binary format.

5. XML is a Family of Technologies

"The XML family" is a growing set of interrelated languages and modules that offer useful services to accomplish important and frequently demanded tasks:

- Xlink: to add hyperlinks to an XML file.
- XPointer and XFragments: to point to parts of an XML document.
- Cascading Style Sheets (CSS): the style sheet language applicable to XML as it is to HTML.
- Extensible Stylesheet Language (XSL): based on XSLT to rearrange, add, and delete tags and attributes.
- Document Object Model (DOM): a standard set of function calls for manipulating XML (and HTML) files from a programming language.
- XML Schema: help developers define the data structures of their own XML-based formats.

6. XML is New, But Not That New

Although, the development of XML started in 1996 and has been a W3C Recommendation since February 1998, overall this technology is not very new. Before XML there was SGML, developed in the early '80s, an ISO standard since 1986, and widely used for large documentation projects. XML has taken the best parts of SGML, guided by the experience with HTML, to make structured data more regular and simple to use.

7. XML Leads HTML to XHTML

XHTML is the successor of HTML, and has many of the same elements as HTML. However, the syntax has been changed to conform to the rules of XML.

8. XML is Modular

XML allows the definition of a new document format by combining and reusing other formats that have common elements or attribute names. The XML namespace mechanism is a way to eliminate name confusion when combining formats.

9. XML is The Basis for RDF and The Semantic Web

W3C's Resource Description Framework (RDF) is an XML text format that supports resource description and metadata applications, such as music play-lists, photo collections, and bibliographies. RDF can integrate applications and agents into one Semantic Web that enables computers to communicate effectively.

10. XML is License-free, Platform-independent and Well Supported

XML is open-source technology that permits access to a large and growing community of tools and experienced engineers.

C. EXTENSIBLE 3D (X3D) GRAPHICS

X3D Graphics is the next-generation version of the Virtual Reality Modeling Language 1997 (VRML97) 3D graphics format for the Web. The X3D Graphics Working Group is designing and implementing the X3D specification using XML to equivalently express the geometry and behavior capabilities of VRML 97.

X3D-Edit is an authoring tool enabling developers to create valid scene graphs with little trouble [Brutzman 2001]. The tool is simple enough that even novice 3D developers can create virtual worlds with relative ease. When generating a scene, the X3D-Edit software limits the developer to select only allowable choices for node elements, eliminating guesswork and potential errors. X3D-Edit utilizes IBM's Xena XML editor configured to work with the X3D document type definition (DTD). Documents are translated into VRML using an XSLT stylesheet, and a 3D-capable Web browser is automatically launched for convenient debugging. [Brutzman 2001] Figure II.1 shows a screen Capture of the X3D-Edit tool.

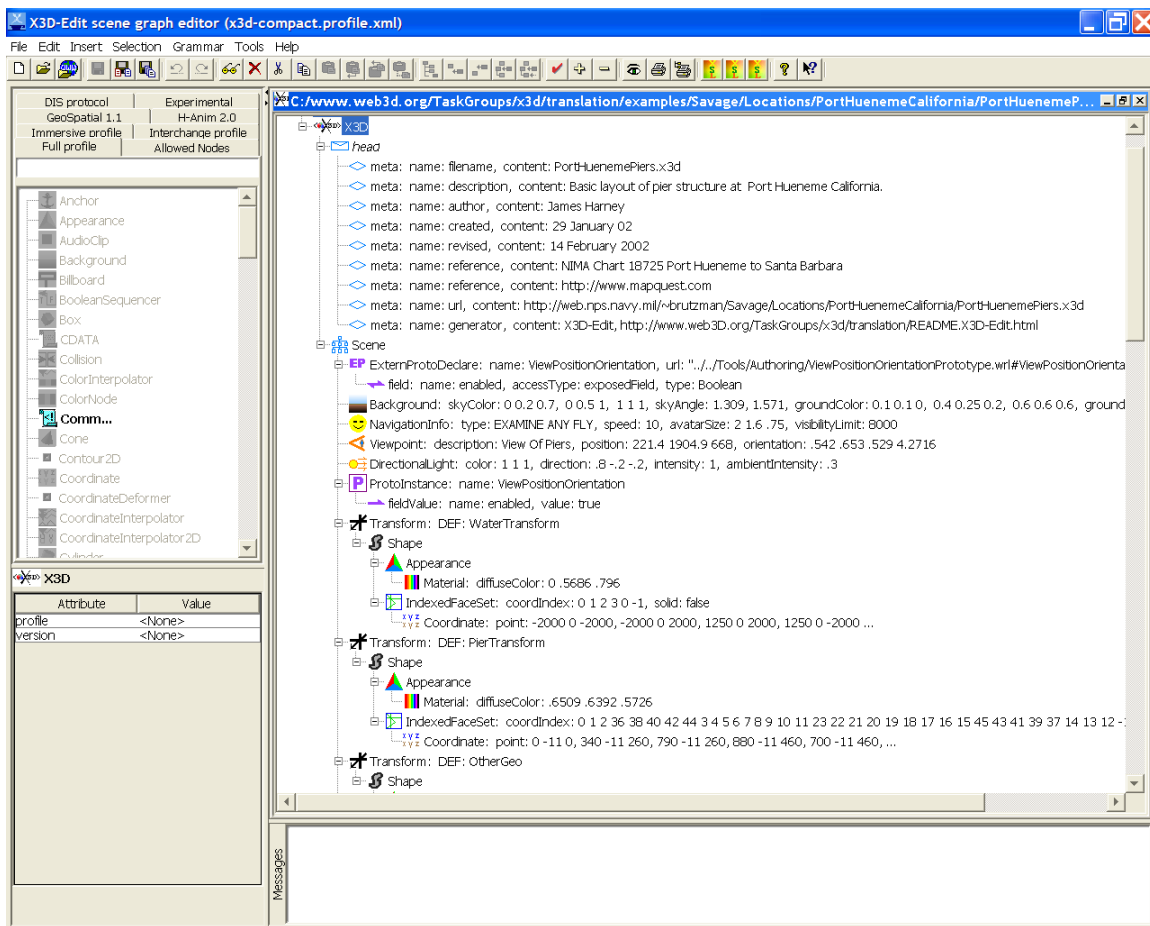


Figure II.1 Screen Capture of the X3D-Edit Tool

D. MAJ. CLIF WILLIAMS' THESIS: HTTP/XML Serving

In Maj. Williams' thesis [Williams 2000], the objective was to demonstrate and to evaluate the functionality of an XML-based Web framework, which allows for the separation of application programming and Web presentation. This Web framework research demonstrated the ability to connect to both XML and non-XML heterogeneous databases and provide a means for distributing data across heterogeneous systems.

This XML-based Web application framework provides a framework for distributing and visualizing 3D simulations utilizing X3D and VRML. Specifically, it illustrates the development of an XML-based Application Server that supports the use of multiple XSLT stylesheets to enable translations between XML information and 3D VRML scene representations.

The specific approach demonstrated in this thesis is to use an XML Web application server to provide systems interoperability, then to develop XSL stylesheets to enable automated translations between XML and users' desired system outputs. Cocoon was selected for the following reasons:

- It has demonstrated consistent success within the Java and XML community.
- It is an XML Application Server-based program, which allows any conformant parser to be used.
- It is based on the Java Servlet technology.
- It is part of the Apache XML project, and has default support for Apache Xerces and Apache XALAN.

Further details on Cocoon are explained in Chapter III.

E. MAJ. SHANE NICKLAUS' THESIS: XML OPORDER VISUALIZATION

X3D and VRML graphics can provide a complete capability for scalably visualizing an amphibious operation in a 3D battlespace. MAJ. Shane Nicklaus' thesis aimed to present an innovative, interoperable methodology for communicating and visualizing tactical orders. Particularly, the thesis showed to translate and synthesize United States Message Text Format (USMTF) based Operation Order data messages written in XML into a 3D amphibious raid within a virtual battlespace. By combining the power of XML and VRML, the result was a 3D world populated with aircraft, ships, amphibious vehicles, and both land and sea terrain. Such virtual worlds permit the viewer to navigate through an amphibious operation in both space and time, and thus observe the interactions between objects in that world. In fact, this thesis demonstrated and evaluated the functionality of 3D amphibious operation visualization using X3D and VRML. [Nicklaus 2001].

F. LT. JAMES HARNEY'S THESIS: AGENT-BASED ANTI-TERRORISM/FORCE PROTECTION (AT/FP) MODEL

For this thesis an Anti-Terrorism and Force Protection for Navy ships simulation model is developed as an exemplar of support system warfare. All work in this research was conducted in a web-base, 'user-centric' fashion, utilizing a combination of user and agent-based control of entities for simulation iterations, plus various Open Source technologies that include X3D, Scalable Vector Graphics (SVG), and XML. This work is a fully integrated prototypical Java-based application proving the efficient application of various Open-Source and Web-based technologies that can to provide the tactical operator with tools to aid in Force Protection planning. This thesis demonstrates and evaluates the functionality and effectiveness of employing web-based modeling and simulation technologies such as X3D Graphics and agent-based simulation towards the defensive planning of Anti-Terrorist and Force Protection measures against the surface-borne terrorist threat. [HARN03]. The AT/FP thesis served as the primary user of the work produced by this thesis.

G. HTTP AND APACHE WEB SERVER XML SERVING

HTTP was originally conceived as a simple, lightweight request-response protocol for communication between web browsers and servers. An HTTP server supports a small number of simple request actions (referred to as "methods"), e.g. GET, POST, and HEAD. The response "document" returned by the server can be encoded in any standard MIME type, including generic types for text or unidentified binary content. Within this general framework, developers and vendors have invented a wide variety of schemes for encoding application data within the request message body and the response.

XML can provide a partial solution to the standardization of content formats. XML formatted data may be inserted into the body of an HTTP POST request to include relatively complex data that is both human and machine-readable. The server response can also include an XML document containing the results of whatever was requested. [Carlson 1998].

Apache is open-source HTTP server software, implemented for several for modern operating systems including UNIX and Windows. The main goal of the Apache project is to provide a secure, efficient and extensible server that provides HTTP services consistently with the current HTTP standards. Actually, Apache has been the most popular web server on the Internet since April of 1996. The August 2002 Netcraft Web Server Survey found that 63% of the Web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined. To learn more about this project and to download and install Apache, see the main Apache web site <http://httpd.apache.org>.

For the purpose of this thesis, we have downloaded and installed Apache 2 from <http://www.apache.org> to an NPS server <http://terra.cs.nps.navy.mil>.

H. SUMMARY

This chapter describes thesis as a related work and globally defines the basic technologies exploited in order to provide an understanding for the thesis. The reader may refer to the List of References for more details.

III. EXAMPLE MILITARY TACTICAL DATABASES

A. INTRODUCTION

This chapter examines different kinds of databases, and presents examples of databases containing military tactical parameters. Such comparisons are useful for defining entity specifications employed in the AT/FP scenario.

B. PARAMETERS OF INTEREST INCLUDING ORIGIN AND METADATA

A *database* is one or more large structured sets of persistent data, usually associated with software to query and update the data. A simple database might be a single file containing many records, each of which contains the same set of fields where each field is a certain fixed width. Alternatively, a database may contain multiple dissimilar tables of records.

1. Compare Static to Dynamic Database Origin

A static database consists of "information-based relationships" is rigorously structured to facilitate retrieval and update in terms of inherent relationships. This creates a static environment wherein the locations of the related records are already known. Typical static Database Management Systems (DBMSs) are either hierarchical or a CODACYL (network or plex) DBMS. These environments facilitate rapid, high-volume processing of complex dissociated data.

However, a dynamic database is the one with "value-based" relationships where typically, the relationship is specified at retrieval time and the locations of related records are discovered during retrieval. Both Independent Logical File (ILF) databases and relational databases are value-based. [Denis 1998]

In terms of military simulation, static database is typically composed of entities such as tank, submarine, carrier, fighter aircraft, missiles, bridges, or other elements. Each entity has its own specifications (e.g. capacity, speed, volume, etc.). Furthermore, there is an operational environment surrounding these entities. This environment

includes terrain, atmospheric, and oceanographic information. Such databases tend to be regular and static, with predictable structure that remains consistent over time.

C. NATO BATTLESPACE GENERIC HUB (BGH)

A common specification for data structures is the first requirement for achieving automated information exchange. XML provides this capability well. The second need is to populate databases and data structures with tactical information. The U.S. Naval Institute (USNI) database, the Federation of American Scientists (FAS) Web site, and other data sources make this possible, although terminology can vary greatly. Thus, a common vocabulary with consistent semantic meanings is the necessary third requirement for information exchange.

Formally known as the Battlespace Generic Hub (BGH), the NATO Land C2 Information Exchange Data Model (LC2IEDM) is the first information exchange model designed to be sufficiently flexible and generic to satisfy the change over time of information exchange requirements (IERs). NATO nations require the LC2IEDM as a minimum to preserve the meaning and relationships of the information exchanged.

The main goal of the Battlespace Generic Hub model is to serve as a common information-exchange data model. The model provides the basis for sharable structured information. It defines standard elements of information (e.g. vehicle AAV, or Location Grid 543789) and can capture relationships between the information (e.g., AAV is located at Grid 543789). (NATO, 2002)

Three models make up the BGH LC2IEDM:

- *Conceptual Data Model:* This model is considered the high-level structures of information; it represents the general concept of information such as Action, Organization, Personnel, Features, Location, etc.
- *Logical Data Model:* This model specifies the way data are structured with an entity-attribute relationship. For example, a tank is an armored fighting vehicle, which is a piece of equipment, which is piece of materiel.

- *Physical Data Model:* this model provides the necessary detailed specification to define the internally reproducible structure of a database.

The functional areas that define the specific inputs of the primary evolution Battlespace Generic Hub data model are conventional fire support, barrier engineering operations, communications and electronics, and personnel administration.

Figure III.1 illustrates the interdependence concept of the Generic Hub and the functional areas. [NATO, 2002]

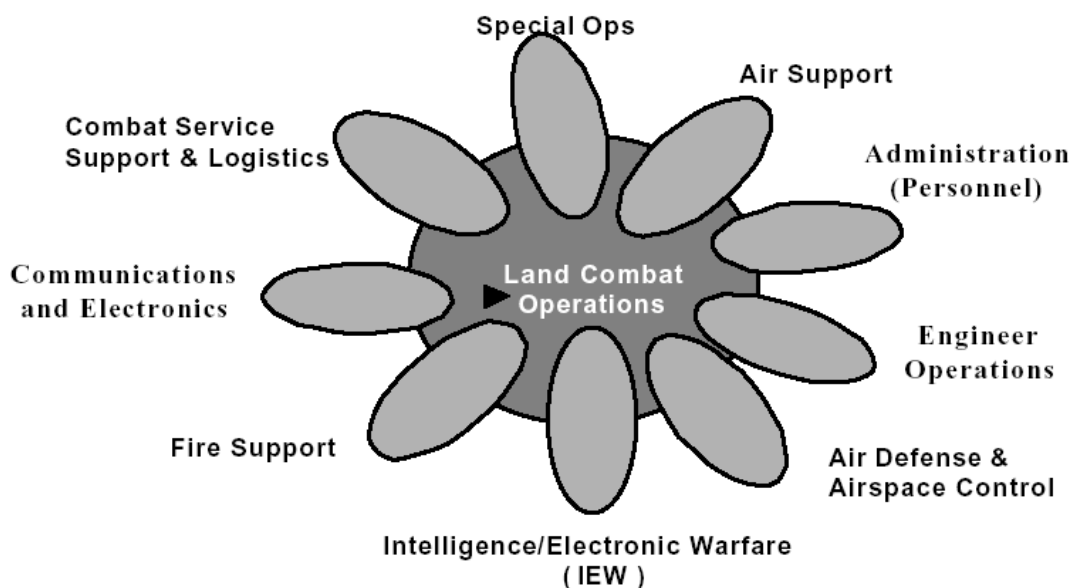


Figure III.1 Generic Hub and Its Relationship to Subfunctional Areas
(From NATO, 2000)

D. TACTICAL DATA SOURCES

1. FAS and USNI

- Capabilities, limitations, and availability

The Federation of American Scientists (FAS) is a nonprofit, tax-exempt, 501c3 organization founded in 1945 as the Federation of Atomic Scientists. The founders of FAS were originally members of the Manhattan Project, creators of the atom bomb and deeply concerned about the implication of its use for the future of humankind.

Known in its early years as the "scientists lobby," FAS combines the scholarly

resources of its member scientists and informed citizens with knowledge of practical politics. Endorsed by the nearly 60 Nobel Laureates in biology, chemistry, economics, medicine, and physics as sponsors, FAS is uniquely qualified to bring the scientific perspective to public policy including weapons nonproliferation and information technologies. FAS strategies include advocacy, briefings with policy makers and the press, public education and outreach, collaboration with civil rights, human rights, and arms-control groups, and grassroots organizing.

FAS is considered one of the best unclassified public resources providing a database of military models. For example in its website <http://www.fas.org>, FAS provides unclassified pictorial representations of various military vessels with the necessary specifications that are sufficient to implement realistic 3D models. Figure III.2 shows one example of vessel “DDG-51, Arleigh Burke class destroyer” downloaded from FAS website to create a 3D model of DDG51 for use in the ATFP scenario. Figure III.3 shows the corresponding 3D model created by LT Harney using X3D. [Harney 2003]

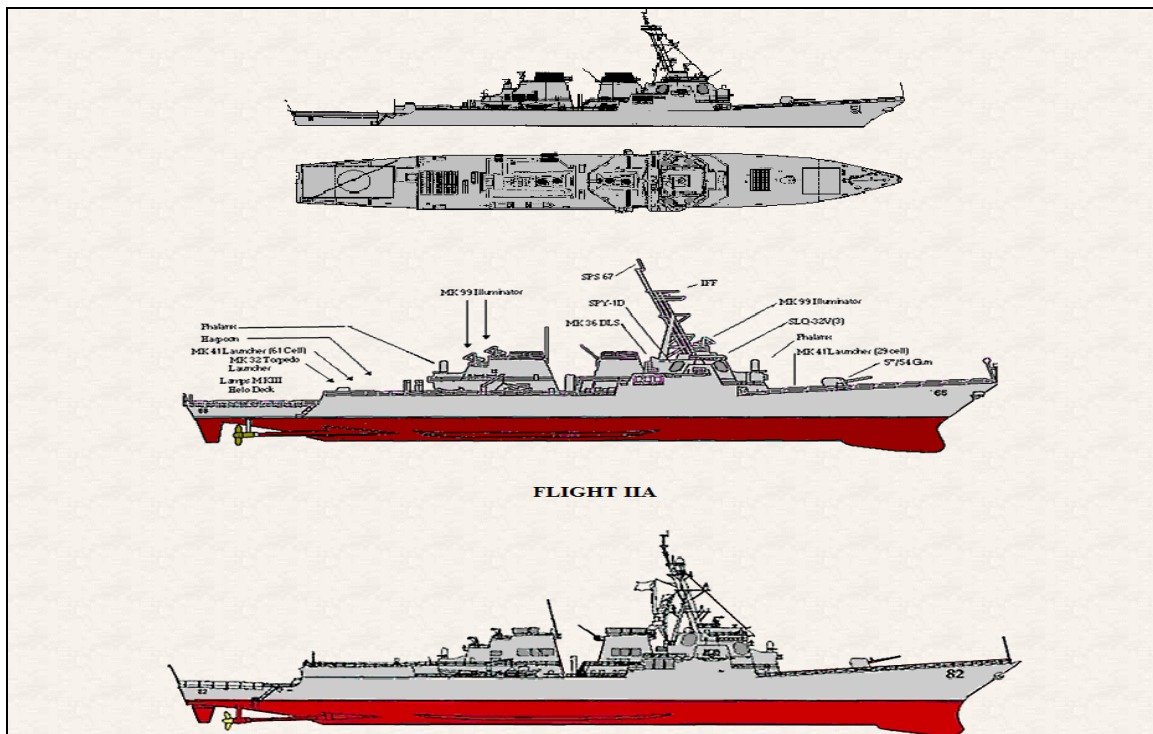


Figure III.2. Depiction of DDG-51, Arleigh Burke class destroyer from the fas.org web site <http://www.fas.org/man/dod-101/sys/ship/ddg-51.htm>

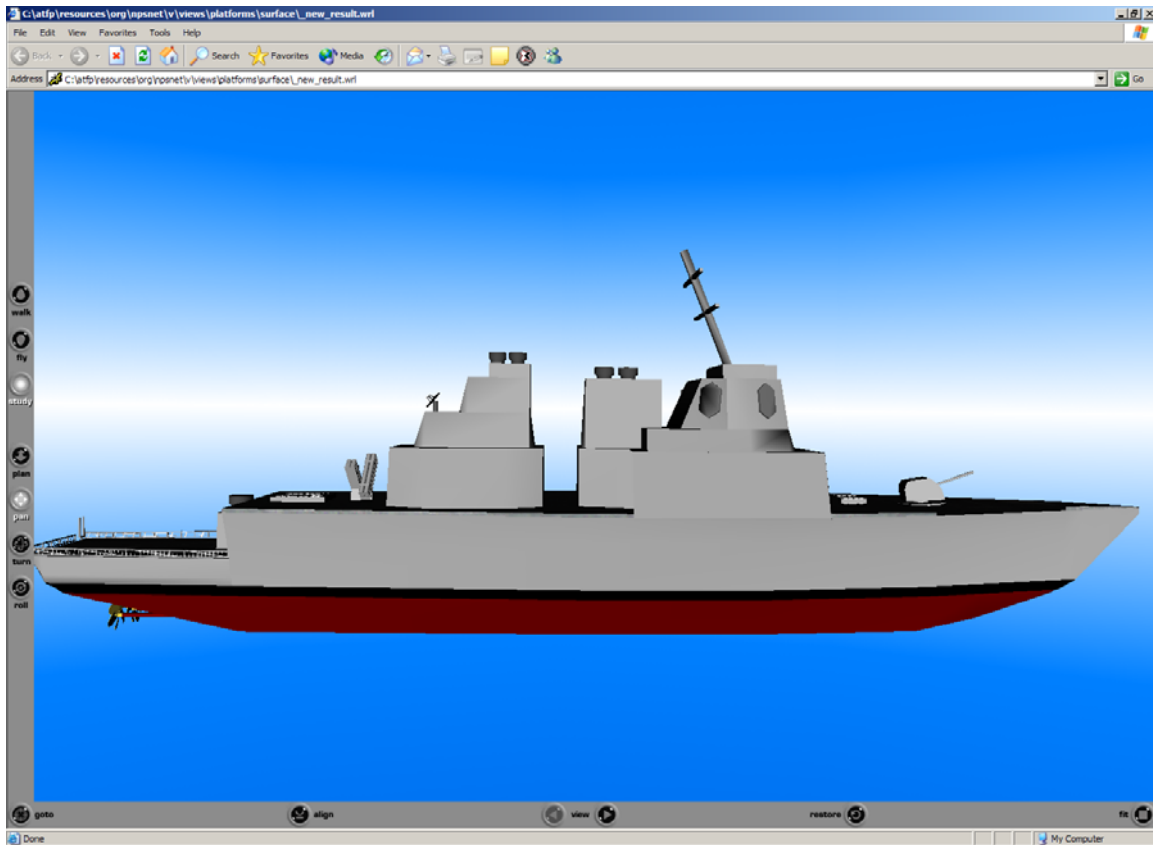


Figure III.3. Screen Snapshot of the Arleigh Burke Class X3D Model. [FromHarney 2003]

U.S. Naval Institute (USNI) is another effective source of military information. USNI has the mission to contribute to the nation's security by providing an open forum where ideas and issues important to the Sea Services in particular (and the other military services in general) can be advanced.

The Naval Institute provides maritime education and awareness internationally, at every level, from high school to college and beyond, for professionals and policymakers, while at the same time providing for the needs of serving professionals in the sea services. In its website <http://www.usni.org>, 3D modelers may find images and specification of components needed to create military models.

2. Rolands & Associates (R & A)

ROLANDS & ASSOCIATES (R&A) Corporation was originally established in 1982 with a focus on the design, development and support of modeling and simulation (M&S) applications. One of the projects developed by R&A is "Support for the NATO

Consultation, Command, Control, Agency (NC3A) Computer Aided Exercise (CAX) Tested.”

In this referenced project, R&A provided analytic and software programming support to the NC3A, The Hague. R&A engineers and programmers integrated Join Theater Level Simulation (JTLS) into the Operational Environment Simulation (OES) of the NC3A CAX Testbed. The objective of the NC3A CAX testbed was to provide empirical insight into the issues concerning the implementation of the future ACE CAX capability, which might:

- support adjacent echelon exercises for NATO command echelons and the highest national command echelons;
- be interfaced with the ACE C3I System;
- allow various command echelons in the exercise to use their operational C3I system to perform in the exercise yet remain in their operational command locations;
- provide the capability to integrate different national simulation models to support multinational forces exercises and make the most effective use of existing models.

The JTLS database appears to be well suited for more general adaptation

The web site <http://www.rolands.com> describes selected additional projects of the R&A company.

3. DIS Protocol Enumeration Values

The Distributed Interactive Simulation (DIS) protocol is a government/industry Networked-behavior Communication protocol that defines an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly interactive entities. The primary product of these efforts was a set of networking protocols standardized by the Institute for Electrical & Electronic Engineers (IEEE) [IEEE 1278.x]

In order for DIS to take advantage of currently installed and future simulations developed by different organizations, a means had to be found for assuring interoperability between dissimilar simulations. These means were developed in the form of industry consensus standards. The open forum (including government, industry, and academia) chosen for developing these standards was a series of semi-annual Workshops on Standards for the Interoperability of Distributed Simulations that began in 1989. The results of the workshops have been several IEEE (Institute of Electrical and Electronics Engineers) standards. These standards provide application protocol and communication service standards to support DIS interoperability. In addition, IEEE P1278.3, Recommended Practice for Exercise Management and Feedback, provides user guidelines for setting up and conducting a DIS exercise.

The relationship between the component documents comprising the set of IEEE DIS documents is shown in the figure below. Used together, these standards and associated recommended practice will help to ensure an interoperable simulated environment.

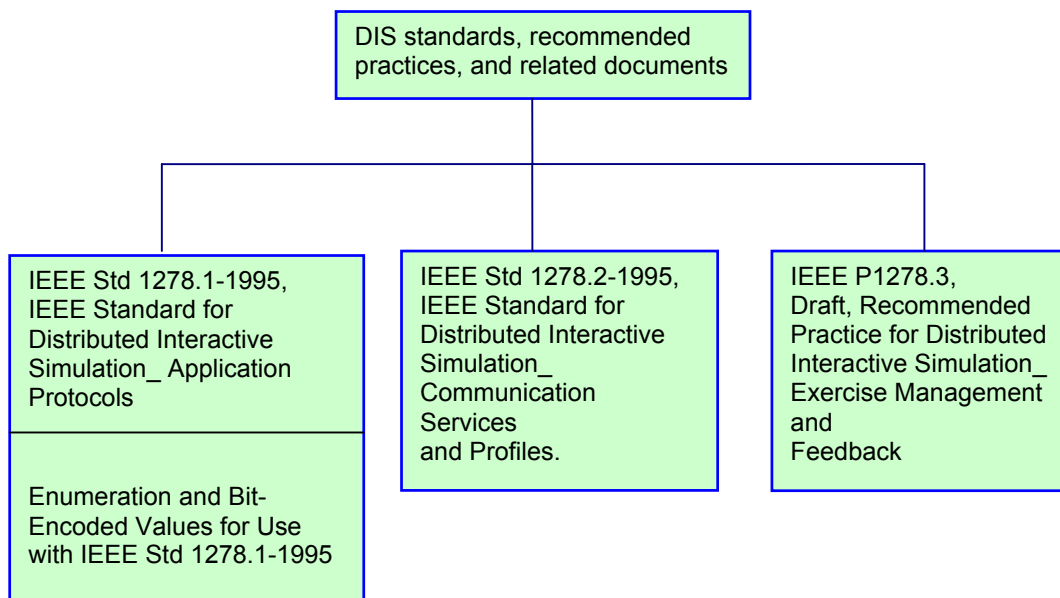


Figure III.4. The relationship between the component documents comprising the set of IEEE DIS documents.

IEEE Std 1278.1-1995 defines both the format and semantics of data messages, also known as Protocol Data Units (PDUs) that are exchanged between simulation applications and simulation management. The PDUs provide information concerning simulated entity states, the type of entity interactions that take place in a DIS exercise, and data for management and control of a DIS exercise. IEEE Std 1278.1-1995 also specifies the communication services to be used with each of the PDUs.

An additional non-IEEE document is required for thorough use of protocol messages with IEEE Std 1278.1-1995. This document is entitled *Enumeration and Bit-encoded Values for use with IEEE Std 1278.1*, and is available from the Tactical Warfare Simulation and Technology Information Analysis Center at the Institute for Simulation and Training of the University of Central Florida. <http://www.sisostds.org>,

IEEE Std 1278.2-1995 defines the communication services required to support the message exchange described in IEEE Std 1278.1-1995. In addition, IEEE Std 1278.2-1995 provides several communication profiles that meet the specified communications requirements.

Together IEEE Std 1278.1-1995 and IEEE Std 1278.2-1995 provide the necessary information exchange for the communications element of DIS.

P1278.3, provides guidelines for establishing a DIS exercise, managing the exercise, and providing proper feedback. When approved, this recommended practice will be used in conjunction with

4. Military Symbology Library

Defining a standard set of rules for military symbol construction is becoming a necessity to promote interoperability at the information-display level. The rules in this standard are considered to be the minimum necessary to ensure that information about warfighting symbology is exchanged successfully across service and organizational boundaries.

This standard is supposed to provide common warfighting symbology along to ensure the compatibility, and (to the greatest extent possible), the interoperability of DoD Command, Control, Communications, Computer, and Intelligence (C4I) systems during

development, operations, and training. These symbols are designed to enhance DoD's joint warfighting interoperability by providing a standard set of common C4I symbols.

The display of warfighting symbology has evolved from a static, manual operation to include fully automated computer generation. [DEFENSE 1999] warfighting symbology standardization incorporates MIL-STD-2525B, *Common Warfighting Symbology*, a DoD symbol data repository, and supporting documentation such as the Symbology Information Technology Standards Management Plan (SITSMP), Configuration Management Plan, and Symbology Standards Management Committee (SSMC) charter (see Ffigure III.4).

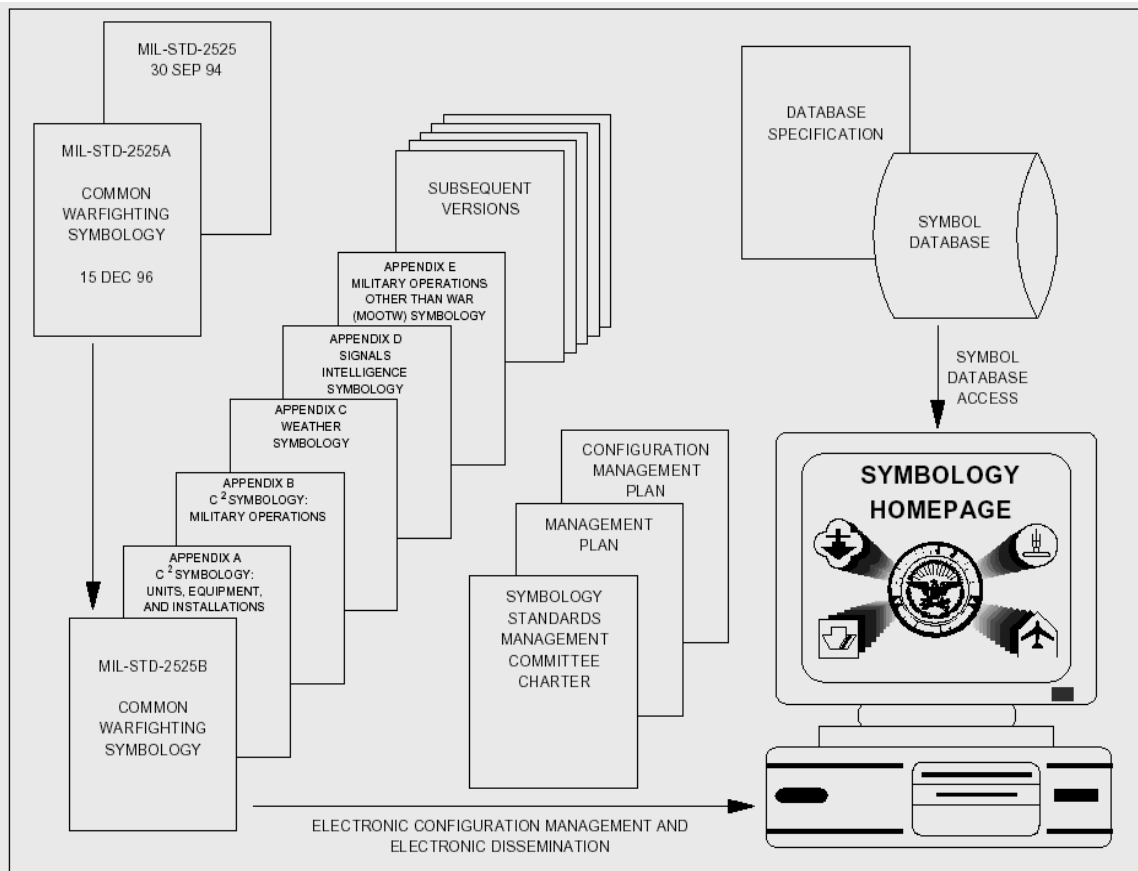


Figure III.5. Common warfighting symbology documents from [From DEFENSE 1999]

E. CONVERTING DATASETS AND DATABASES TO XML

1. XML versus Databases

The first question that a designer needs to ask when thinking about XML and databases is why we a particular database is needed in the first place. Is there legacy data that must be exposed? Are stories data needed for dynamic Web pages? Is the database needed by an e-commerce application in which XML is used as a data transport? What is the structure at the underlying data? The answers to these questions will strongly influence the choice of database and middleware (if any), as well as how the database is assessed.

For example, suppose there is an application that uses XML as a data transport. Such data will have a highly regular structure and can be used by non-XML applications. Furthermore, details like the specific names of entities and the encodings used by XML documents probably are not important-- after all, the interest is in the data containing, not how it is stored in an XML document. In this case, a relational database software to transfer the data between XML documents and the database is sufficient. If the applications are object-oriented, a system that can both store those objects in the database and serialize them as XML might be preferable.

On the other hand, suppose a Web site is built from a number of prose-oriented XML documents. Managing the site and providing a way for users to search its contents are the basic operations that to be done. Such documents are likely to have a less regular structure, so details such as entity usage are probably important because they are a fundamental part of how documents are structured. In this case, a native-XML database or a content management system is much preferable. This will allow preserving physical document structure, supporting document-level transactions, and executing queries in an XML query language. [Bourret 2003]

2. XML from Databases and Vice Versa

Many different open-source or commercial software tools can be used to transfer data between XML documents and databases.

- *Example 1: XML Spy*

XML Spy is one of the basic tools in this research. This tool is described in detail in Chapter IV.

<p>Developer: Altova</p> <p>URL: http://www.xmlspy.com/features_convert.html</p> <p>License: Commercial</p> <p>Database type: Relational (ADO)</p> <p>Direction(s) : Database=>XML, XML=>Database</p> <p>Entry last updated: April, 2002</p>
--

XML Spy can transfer (export) the result of a SELECT statement to an XML document using a table model. The export function has a number of options, such as whether to transfer column data to attributes or child elements, whether to exclude primary and foreign key values (useful when these are database-specific, rather than containing business values), and what format to use for dates and numbers. XML Spy can also transfer (import) data from an XML document to the database, apparently using an object-relational model. The import function can optionally generate key values and the user can specify both the element to start with as well as the number of levels of nesting to consider. XML Spy can also optionally create tables in the database as needed. The import and export functions are programmatically addressable through a software API.

- *Example 2: XVRL (Extensible Value Resolution Language)*

<p>Developer: xvrl.org</p> <p>URL: http://www.xvrl.org</p> <p>License: Open Source</p> <p>Database type: Relational (JDBC)</p> <p>Direction(s) : Database=>XML</p> <p>Entry last updated: Summer, 2001</p>

XVRL is a template language that allows programmers to retrieve data from external sources, including relational databases. The language includes elements to declare a "resolver" class and then invoke methods on it. The results of these method calls are returned to the processing software, which embeds them in the output document.

It appears that virtually any Java class can be used as a "resolver" class. Thus, the product is arbitrarily extensible. The software comes with a `SQLDatabaseResolver` class, which can return the results of a `SELECT` statement as XML using a table-based mapping.

There are two ways to process an XVRL document. First, `JResolver` is a simple command line utility that accepts an XVRL document and an optional XSLT stylesheet and returns the processed document. Second, `JAVR` (Java API for Value Resolution) allows you to process XVRL documents in your own application. `JAVR` conforms to Apache's Transformation API for XML (TrAX).

Table III.1 provides a list of current products enabling the conversion between XML and Databases. The source of this table and the other tables related to this chapter is at <http://www.rpbouret.com/xml/XMLDatabaseProds.htm>

Table III.1 List of products enabling the conversion between XML and Databases
[From Bourret 2003]

Product	Developer	License	DB Type	DB=>XML	XML=>DB
ADO	Microsoft	Commercial	Relational	x	X
Allora	HiT Software	Commercial	Relational	x	X
ASP2XML	Stonebroom	Commercial	Relational	x	X
Attunity Connect	Attunity Ltd.	Commercial	Relational, hierarchical, etc.	x	X
Beanstalk	Beanstalk Corp.	Commercial	Relational	x	--
Castor	exolab.org	Open Source	Relational	x	X
Charteris Integration Toolkit	Charteris	Commercial	Relational	x	X
DatabaseDom	IBM	Evaluation only	Relational	x	X

Data Junction	Data Junction, Inc.	Commercial	Relational, ISAM, etc.	x	X
DB2XML	Volker Turau	Open Source	Relational	x	--
DBIx::XML_RDB	Matt Sergeant	Open Source	Relational	x	X
DBIx::XMLMessage	Andrei Nossov	Open Source	Relational	x	X
DB-X	Swift Inc.	Commercial	Relational	x	X
DB/XML Transform, DB/XML Vision	DataMirror Corp.	Commercial	Relational	x	X
Delphi	Borland	Commercial	Relational	x	X
Easysoft XML-ODBC Server	Easysoft	Commercial	Relational	x	--
e.Reporting	Actuate	Commercial	Relational	x	--
eTools XML	GA eXpress	Commercial	MultiValue	x	X
iConnector	infoteria	Commercial	Relational, Lotus Notes	x	X
idx-xmnesia	IDEALX	Open Source	mnnesia	x	X
iINTEGRATOR	Lightspeed Interactive	Commercial	Relational	x	x(?)
Import/Export Studio	Etasoft	Commercial	Relational	x	X
InterAccess	XML Software Corporation	Commercial	Relational	x	X
JaxMe	Jochen Wiedmann	Open Source	Relational	x	X
jXTransformer	DataDirect Technologies	Commercial	Relational	x	X
ODBC2XML	Intelligent Systems Research	Shareware	Relational	x	--
ODBC Socket Server	Team FXML	Open Source	Relational	x	--
Osage	George Stewart, et al	Open Source	Relational	x	X
PowerXML Pro	Wizen Software	Commercial	Relational	x	x(?)
PXSLServlet	Paul A.	Open Source	Relational	x	X

	Tchistopolskii				
sql2dtd	David Mertz	Public domain	Relational	x	--
sql2xml	Scott Hathaway	Public domain	Relational	x	--
Sunopsis XML Driver	Sunopsis	Commercial	Relational	x	X
SXQL	Goetz Hatop	Shareware	Relational	x	X
Sypram XML Studio	Sypram Technology	Commercial	Relational	x	--
TransVerse	Coyote Consultants	Commercial	Relational, MultiValued	x	X
XChainJ	Cogent Logic Corp.	Commercial	Relational	x	X
X:Forge	Bibop Research International	Open Source	Relational, Native XML	x	--
xlinkit	xlinkit.com	Free/commercial	Relational	x	--
XML DataDesk	NetBryx Technologies	Commercial	Relational	x	--
XMLDataLink	BASD, Integra Micro Systems (P), Ltd.	Commercial	Relational	x	x(?)
XML-DBMS	Ronald Bourret, et al	Open Source	Relational	x	X
XML::Generator::DBI	Matt Sergeant	Open Source	Relational	x	--
XMLizer	E-XMLMedia	Commercial	Relational	x	X
XML Junction	Data junction, Inc.	Commercial	Relational, ISAM, etc.	x	X
XML Lightweight Extractor (XLE)	IBM	Evaluation	Relational	x	--
XML Servlet	Cerium Component Software Incorporated	Commercial	Relational	x	X
XMLShark	infoShark	Commercial	Oracle	x	X
XML Spy	Altova	Commercial	Relational	x	X
XML SOL	Oracle	Development-	Relational	x	X

Utility for Java		only			
XML Transformation Server (XTS)	iWay Software	Commercial	Relational, hierarchical, etc.	x	X
XSQL Servlet	Oracle	Development-only	Relational	x	--
XVRL (eXtensible Value Resolution Language)	xvrl.org	Open Source	Relational	x	--

3. XML-Enabled Databases

They are Databases with extensions for transferring data between XML documents and themselves. Table III.2 contains a current list of these kinds of databases.

Table III.2 List of XML-Enabled Databases [From Bourret 2003]

Product	Developer	License	DB Type
Access 2002	Microsoft	Commercial	Relational
Cache	InterSystems Corp.	Commercial	Multi-valued
DB2	IBM	Commercial	Relational
eXtremeDB	McObject	Commercial	Navigational
FileMaker	FileMaker	Commercial	FileMaker
FoxPro	Microsoft	Commercial	Relational
Informix	IBM	Commercial	Relational
Objectivity/DB	Objectivity	Commercial	Object-oriented
Oracle 8i, 9i	Oracle	Commercial	Relational
SQL Server 2000	Microsoft	Commercial	Relational
Sybase ASE 12.5	Sybase	Commercial	Relational
Versant enJin	Versant Corp.	Commercial	Object-oriented

4. XML Servers

XML-aware J2EE servers, Web application servers, integration engines, and custom servers are different categories of XML Servers. Some of these are used to build distributed applications while others are used simply to publish XML documents to the Web.

- Example: Cocoon

Developer: Apache Software Foundation

URL: <http://xml.apache.org/cocoon>
<http://xml.apache.org/cocoon/userdocs/xsp/esql.html>

License: Open Source

Database type: Relational (JDBC)

Entry last updated: February, 2002

Cocoon is an XML publishing framework for the Apache web server. Web pages are written in XSP (Extensible Server Pages), a scripting language wraps Java source code in XML tags. This language contains an SQL tag library (ESQL) for retrieving data from relational databases with JDBC. The tags largely mirror the most commonly used methods in JDBC, such as tags to prepare and execute queries, iterate through rows in a result set, and get column values from a result set. Note is that any SQL statement can be executed, not just a SELECT statement. If the statement is a SELECT statement, individual tags can be used to specify where column values are to be placed in the output document. In the case of joins, a "watch" column can be specified so that the corresponding tag is only created when the column value changes. This allows software to create nested XML in the expected way. Table III.3 provides a current list of XML-servers.

Table III.3. List of XML-Server products [From Bourret 2003]

Product	Developer	License	DB Type
ATG Dynamo	ATG	Commercial	Relational (JDBC)
AxKit	AxKit.com Ltd.	Open Source	Relational
Cocoon	Apache.org	Open Source	Relational
ColdFusion	Macromedia	Commercial	Relational
Enhydra	enhydra.org	Open	Relational

		Source	
Enosys Integration Suite	Enosys Software	Commercial	Relational, XML, HTML, Web services, flat files
Interaction Server	Netegrity	Commercial	Relational
Lasso	Blue World Communications	Commercial	Relational
MatriXML	MojoTechnology, Inc.	Commercial	Oracle. External relational through ODBC
Net.Data	IBM	Commercial	Relational
Nimble Integration Suite	Nimble Technology	Commercial	Relational, IMS, SAP, Notes, flat files, etc.
Noetix Web Query	Aris Software, Inc.	Commercial	Relational
PolarLake Database Integrator	PolarLake	Commercial	Relational (JDBC)
Rhythmyx Content Manager	Percussion Software	Commercial	Relational, Lotus Notes
SQml Web, SQml Publisher	Agave Software Design	Commercial	Relational
Total-e-Business	HP Bluestone	Commercial	Relational (JDBC). Non-relational through Data Source Integration Modules
Visual Net Server	CNet	Commercial	Relational
WebObjects	Apple Computer	Commercial	Relational
XA-Suite	XAware, Inc.	Commercial	Relational, native XML, mainframe, etc.
XML Portal Server (XPS)	Uniway	Commercial	Relational, Lotus Notes, etc.

5. Native XML Databases

As defined by the members of the XML:DB mailing list, a native XML database is one that:

- Defines a logical model for an XML document -- as opposed to the data in that document -- and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of

such models are the XPath data model, the XML Infoset, and the models implied by the DOM and the events in SAX 1.0.

- Has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage.

- Is independent from any particular underlying physical storage model. For example, it can be built on a relational, hierarchical, or object-oriented database, or use a proprietary storage format such as indexed, compressed files.

Native XML databases fall into two broad categories:

Text-based storage. Store the entire document in text form and provide some sort of database functionality in accessing the document. A simple strategy for this might store the document as a Binary Large Object (BLOB) in a relational database, or as a file in a file system, and then provide XML-aware indexes over the document. A more sophisticated strategy might store the document in a customised, optimized data store with indexes, transaction support, and so on.

Model-based storage. Store a binary model of the document (such as the DOM or a variant thereof) in an existing or custom data store. For example, this might map the DOM to relational tables such as Elements, Attributes, Entities or store the DOM in pre-parsed form in a data store written specifically for this task. Model-based storage includes the category formerly known as "Persistent DOM Implementations."

There are two major differences between the two strategies. First, text-based storage can exactly "round-trip" the document, down to such trivialities as whether single or double quotes surround attribute values. Model-based storage can only round-trip documents at the level of the underlying document model. This is typically be adequate for most applications, but applications with special needs in this area need to check to see exactly what the model supports.

The second major difference is speed. Text-based storage obviously has the advantage in returning entire documents or fragments in text form. Model-based storage probably has the advantage in retrieving data quickly, or in combining fragments from different documents. Handling speed does depend on factors such as document size,

parsing speed (for text-based storage), and retrieval speed (for model-based storage). Whether it is faster to return an entire document as a DOM tree or SAX events probably depends on the individual database, again with parsing speed competing against retrieval speed.

Native XML databases differ from XML-enabled databases in three main ways:

- Native XML databases can preserve physical structure (entity usage, CDATA sections, etc.) as well as comments, Processing Instructions (PIs), Data Type Definitions (DTDs), etc. While XML-enabled databases can preserve exact physical structure in theory, this is generally not done in practice.
- Native XML databases can store XML documents without knowing their schema (or DTD), assuming one even exists. Although XML-enabled databases might generate schemas on the fly, this is impractical in practice, especially when dealing with schema-less documents.
- The only interface to the data in native XML databases is XML and related technologies, such as XPath, the DOM, or an XML-specific API, such as the [XML:DB API](#). XML-enabled databases, on the other hand, further offer direct access to the data, such as through ODBC.

For more information about native XML databases, see "Native XML Databases" at <http://www.rpbouret.com/xml/XMLAndDatabases>. Table III.4 provides a current list of commercial and open source native XML Databases.

Table III.4. List of XML Databases [From Bourret 2003]

Product	Developer	License	DB Type
4Suite, 4Suite Server	FourThought	Open Source	Object-oriented
Birdstep RDM XML	Birdstep	Commercial	Object-oriented
Centor Interaction Server	Centor Software Corp.	Commercial	Proprietary

Cerisent XQE	Cerisent	Commercial	Proprietary(?)
Coherity XML Database	Coherity	Commercial	Proprietary
DBDOM	K. Ari Krupnikov	Open Source	Relational
dbXML	dbXML Group	Commercial	Proprietary
DOM-Safe	Ellipsis	Commercial	Proprietary
eXist	Wolfgang Meier	Open Source	Relational
eXtc	M/Gateway Developments Ltd.	Commercial	Multi-valued
eXtensible Information Server (XIS)	eXcelon Corp.	Commercial	Object-oriented (ObjectStore). Relational and other data through Data Junction
GoXML DB	XML Global	Commercial	Proprietary (Text-based)
Infonyte DB	Infonyte	Commercial	Proprietary (Model-based)
Ipedo XML Database	Ipedo	Commercial	Proprietary
Lore	Stanford University	Research	Semi-structured
Lucid XML Data Manager	Ludic'i.t.	Commercial	Proprietary
MindSuite XDB	Wired Minds	Commercial	Object-oriented
Natix	data ex machina	Commercial	File system(?)
Neocore XML Management System	NeoCore	Commercial	Proprietary
ozone	ozone-db.org	Open Source	Object-oriented
Sekaiju / Yggdrasil	Media Fusion	Commercial	Proprietary
Socrates XML	Cincom	Commercial	Object-relational?
Tamino	Software AG	Commercial	Proprietary. Relational through ODBC.
Tendara Mobile XML Database	Tendara	Commercial	Proprietary
TeraText DBS	TeraText Solutions	Commercial	Proprietary
TEXTML Server	IXIA, Inc.	Commercial	Proprietary (Text-based)
TigerLogic XDMS	Raining Data	Commercial	Pick
Virtuoso	OpenLink Software	Commercial	Proprietary. Relational through

			ODBC
XDBM	Matthew Parry, Paul Sokolovsky	Open Source	Proprietary (Model-based)
XDB	ZVON.org	Open Source	Relational (PostgreSQL only?)
Xindice	Apache Software Foundation	Open Source	Proprietary (Model-based)

6. XML Query Engines

XML query engines are standalone programs that can query XML documents. Currently, these implement a variety of languages: XQL, XPath, XML-QL, Quilt, XQuery, etc. The line between XML query engines and native XML databases is somewhat blurry, as native XML databases invariably support an XML query language. In general, XML query engines appear to support querying and nothing else, while native XML databases offer more database functionality: local data store, transaction support, APIs, etc. Avi Rappaport maintains a list of XML query languages and other XML query resources at <http://www.searchtools.com/info/xml-resources.html>. A list of publicly available XQuery implementations on the W3C's XQuery page <http://www.w3.org/XML/Query>

Figure III.6 illustrates the variety of methods available for the conversion of databases to XML documents. [Valikov 2001]

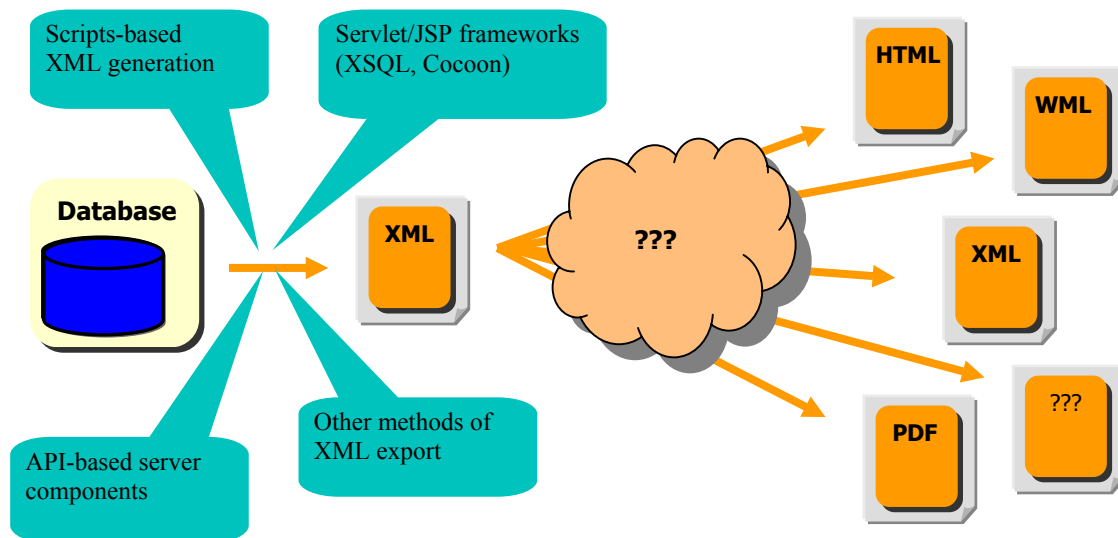


Figure III.6. Methods of converting database to XML [From Valikov2001]

F. DESIGN GOALS AND LESSONS LEARNED

XML is a file format, a way to move information from one machine to another, from one piece of software to another, and a way to transform information to other formats such as XML, HTML, PDF, WML, VRML, etc. The power of XML is growing further because of the universal support it is gaining. In this case, the noise is the substance. Because XML is so low-tech and easy to implement, a company can commit to XML and deliver compatible software within weeks.

Further, XML opens the door for a new generation of databases. Relational databases will not go away, but a new kind of database will take root. In other words, XML is more general than the relational form. Relational databases will only be able to manage a subset of XML runtime. Every developer is going to need an XML workbench, just as every Java developer needs an Integrated Development Environment (IDE) and a Java Virtual Machine(JVM), and a JavaScript programmer needs an editor and a web browser.

G. SUMMARY

This chapter described basic resources that provide a database of military models with tactical information. Such tactical data are useful for defining and populating databases. Therefore, a common information-exchange data model can be provided and constantly updated. A survey of current capabilities and software shows that XML technologies provide many capabilities well by using either open sources or commercial tools.

IV. TOOLS, ARCHITECTURE AND WORK FLOW

A. INTRODUCTION

Many tools are needed for a comprehensive tactical information infrastructure. This chapter defines the principal tools used to perform the representative tasks needed for this research.

B. DEVELOPMENT TOOLS

1. XML SPY Suite

XML Spy is an Altova product and is not an open-source tool. However, NPS has been provided a limited free educational license for students using this tool for their classroom and thesis research work. It is one of the main tools used for this thesis.

XML Spy Suite is a comprehensive and easy-to-use product family that facilitates all aspects of XML Application Development. The product family consists of the XML Spy Document Framework and XML Spy Integrated Development Environment (Spy IDE). The XML Spy Document Framework consists of XSLT Designer and XML Spy Document Editor (IDE).

a. XSLT Designer

XSLT Designer automates the writing of complex XSLT Stylesheets using an intuitive drag-and-drop user interface. It converts XML documents into HTML and automatically creates the XSLT Stylesheets. To accomplish this task, the following steps must be done:

- 1- Load the Schema that forms the basis of an HTML document
- 2- Assign a working XML document which provides preview data
- 3- Drag and drop the specific schema elements into XSLT Designer window
- 4- View the results in the integrated Internet Explorer window

Additionally, XSLT Designer helps to prepare XSLT templates for further use with the Document Editor.

Figure IV.1 shows the three main areas of XSLT Designer:, with the main window at right.

- The top left pane displays a tree view of currently loaded schema file. The schema name and path is displayed in the top line.
- The bottom left pane (the HTML attributes window) consists of several tabs that allow assigning html properties to the elements or attributes in the main window.
- The main window is where the XSLT template must be designed. this window can also view the automatically generated XSLT style sheet and preview the transformation results via Internet Explorer.

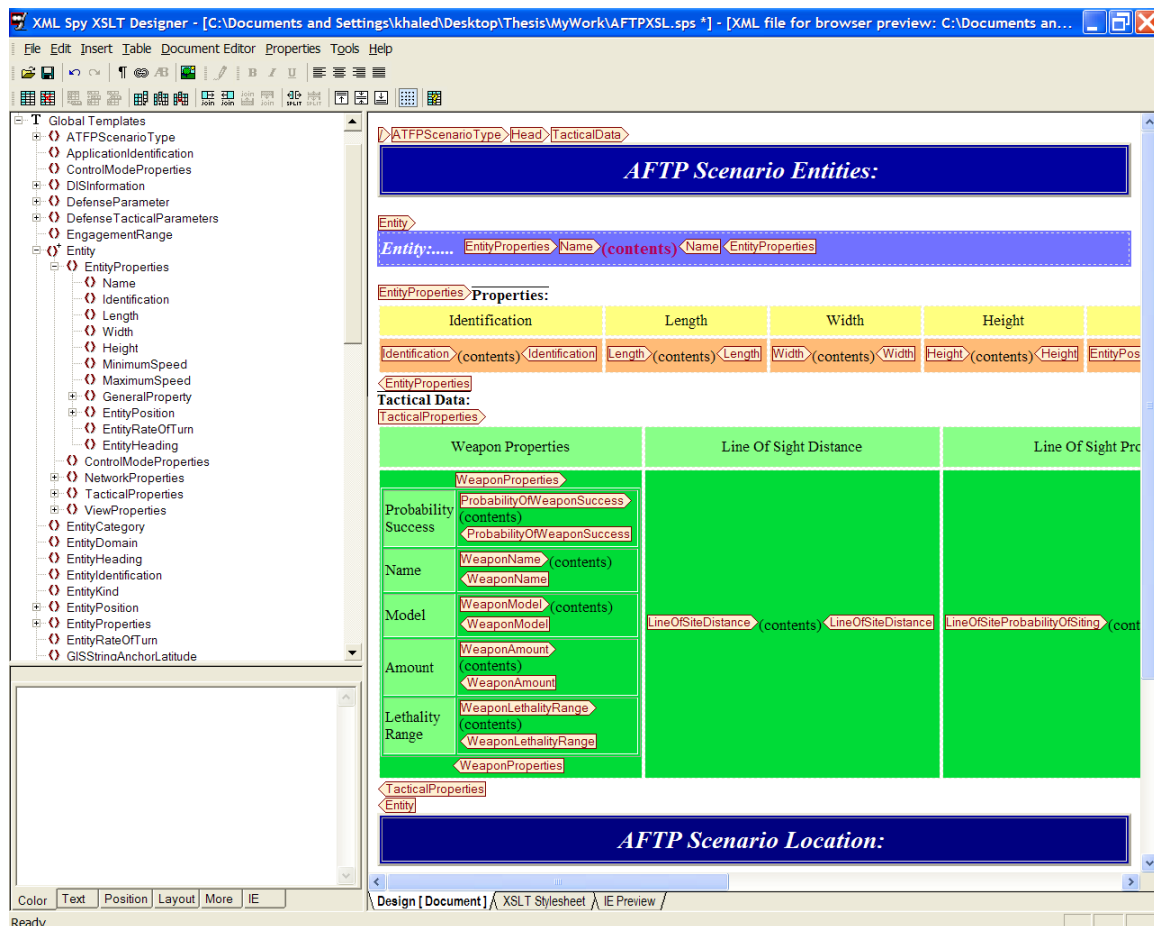


Figure IV.1. Screen Capture showing the three main areas of XML Spy's XSLT Designer.

b. *XML Spy Document Editor*

The XML Spy Document Editor enables users to edit XML documents based on templates created in XSLT Designer. The features of XML Spy IDE are:

- Free-flow “What You See Is What You Get” (WYSIWYG) text editing;
- Form-based data input;
- Presentation and editing of repeating XML elements as tables; and
- Real-time validation and consistency checking using XML Schema

Figure IV.2 is a screen Capture which shows the main window containing a XML document and its corresponding XSL stylesheet. The Browser view icon enables a user to open the Browser view of the XML Document. For the Browser view to display/render an XML document correctly, the XML Document must contain a reference to an XSLT stylesheet, or else an XSLT Stylesheet has to be already assigned to the XML folder or project.

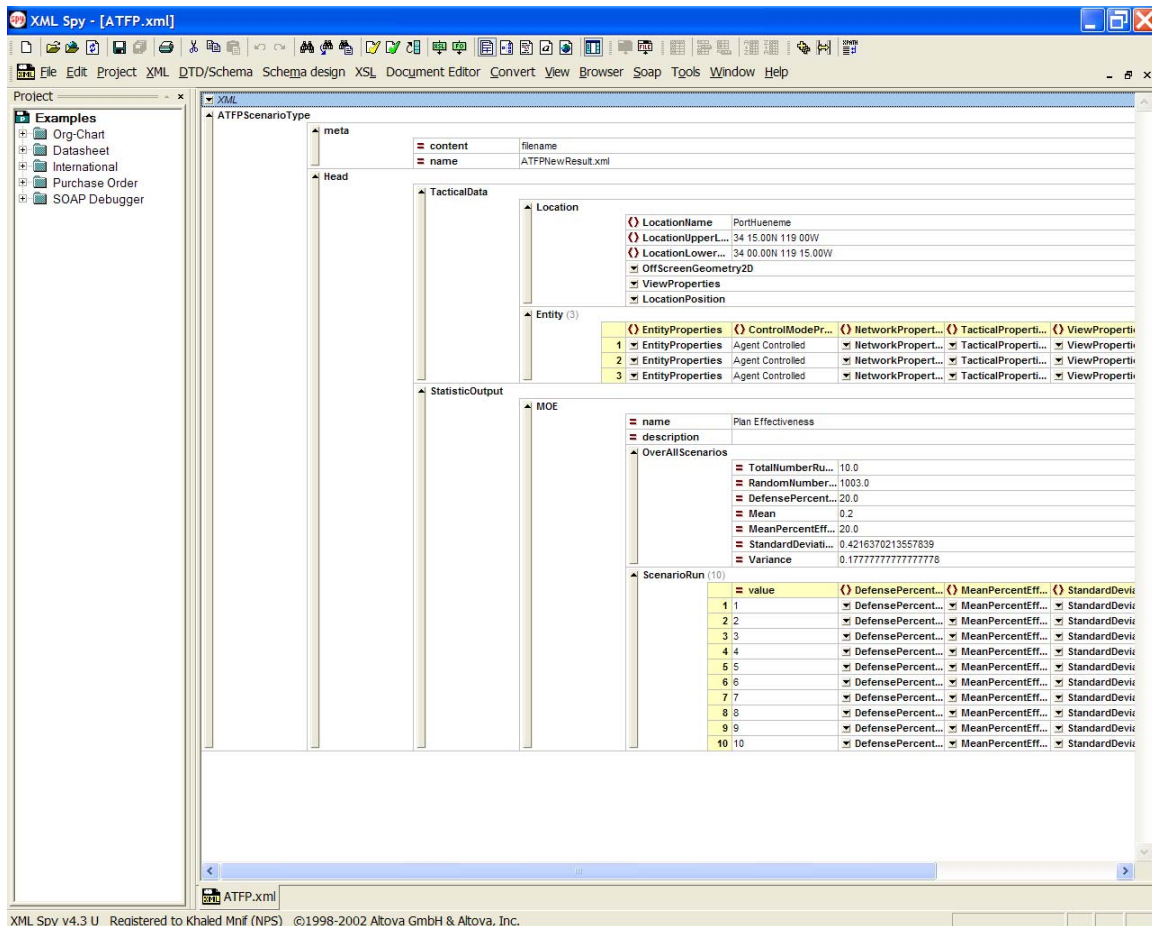


Figure IV.2 Screen Capture of the main window of XML Spy IDE

XML Spy Suite documentation is available on line, and is also available in printed form at <http://www.xmlspy.com/bookstore>. To learn more about XML and various XML-related information resources, there are recommended links at <http://www.xmlspy.com/links>.

2. Saxon XSLT processor

Saxon is an XSLT processor. which is a program that takes both an XML document and a style sheet as inputs and produces a result document as output. The output format may be XML, or HTML, or some other format such as comma separated values, Electronic Data Interchange (EDI) messages, or data in a relational database.

The Saxon package is open-source Java and consists of a collection of tools for processing XML documents. The main components are:

- An XSLT processor, which implements the Version 1.0 XSLT and XPath Recommendations from the World Wide Web Consortium, found at <http://www.w3.org/TR/1999/REC-xslt-19991116> and <http://www.w3.org/TR/1999/REC-xpath-19991116> with a number of powerful extensions.

- A Java library, which supports a similar processing model to XSL, but allows full programming capability, which is needed for performing complex processing of the data or to access external services such as a relational database.

Thus user can invoke Saxon by writing XSLT stylesheets, by writing Java applications, or by any combination of the two. [Kay 01]. For more detailed information, downloading, and installation see <http://saxon.sourceforge.net>. Figure IV.3 shows Saxon architecture, illustrates how it works.

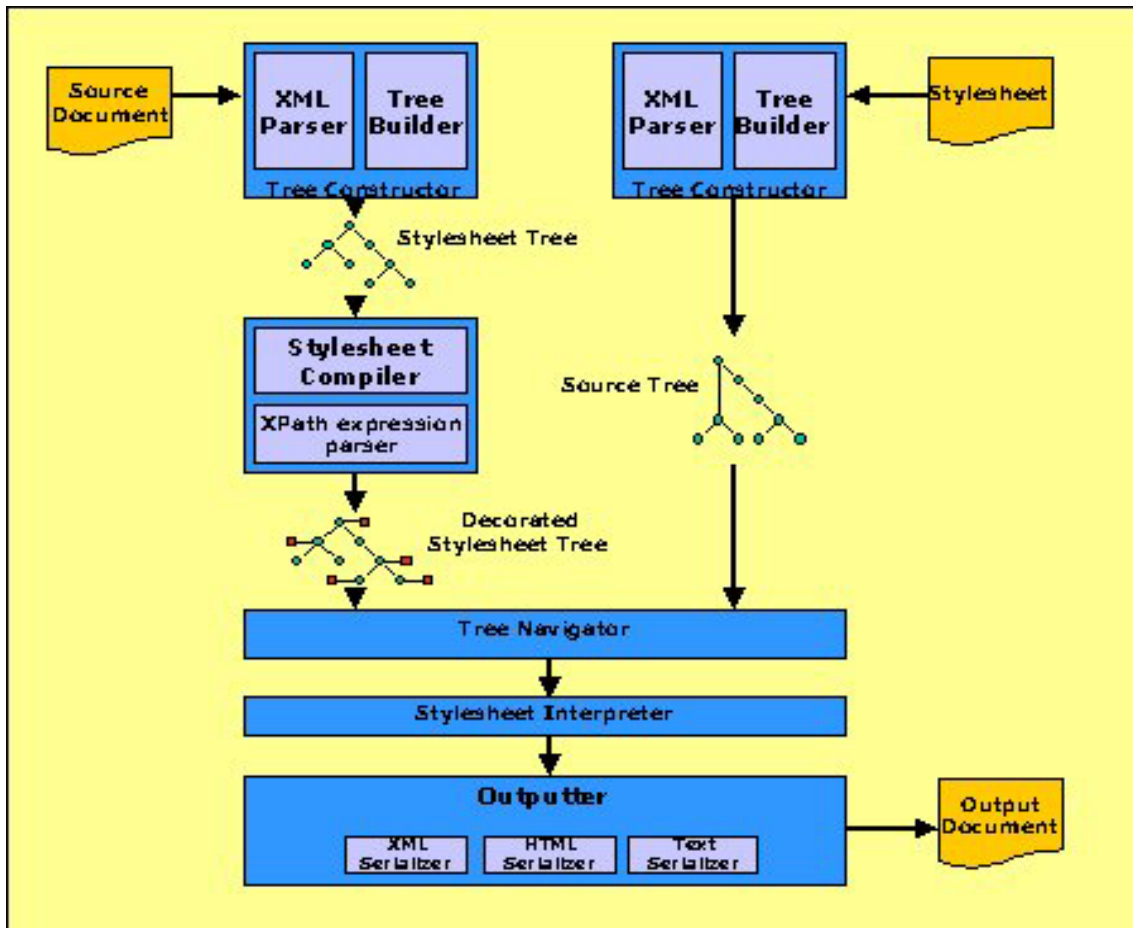


Figure IV.3. Saxon architecture for XSLT processing [From Kay 2001]

The *tree constructor* creates a tree representation of a source XML document. It is used to process both the source document and the style sheet. There are two parts to this:

- The *XML parser* (package `com.icl.saxon.aelfred`) reads the source document and notifies events such as the start and end of an element.
- The *tree builder* (module `com.icl.saxon.Builder`) is notified of these events. It uses them to construct an in-memory representation of the XML document.

The *tree navigator*, as the name suggests, allows applications to select nodes from the tree by navigating through the hierarchy.

The *stylesheet compiler* analyzes the stylesheet prior to execution. It produces a decorated-tree *representation* of the stylesheet in which all XPath expressions are validated and parsed, all cross-references are resolved, stack-frame slots are pre-allocated, and so on. The decorated tree then comes into play at transformation time to drive the style sheet processing. (The compiler is distributed across the classes in the `com.icl.saxon.style` package, especially the methods `prepareAttributes()`, `preprocess()`, and `validate()`).

The *style sheet interpreter* (class `com.icl.saxon.Controller`, which implements the `javax.xml.transform.Transformer` interface in JAXP 1.1) is the core of the Saxon processor. This interpreter uses the decorated stylesheet tree to drive processing. Following the processing model of the language, it first locates the template rule for processing the root node of the input tree. Then it evaluates that template rule (i.e is "instantiated," in the jargon of the standard). [Kay 01]. This approach is very effective, and Saxon is a premiere tool for XSLT.

3. Xerces

Xerces is a sub-project of *Apache XML Project*. It provides world-class XML parsing and generation. Fully-validating parsers are available for both Java and C++ programs. There are two basic approaches to parsing an XML document:

a- The Simple API for XML (SAX) parses an XML document in a sequential manner, generating and throwing events for the application layer to process as it

encounters different XML elements. This sequential approach enables rapid parsing of XML data, especially in the case of long or complex XML documents; however, the downside is that a SAX parser cannot be used to access XML document nodes in a random-access or non-sequential manner.

b- The Document Object Model (DOM) builds a tree representation of the XML document in memory, and then uses built-in methods to navigate through this tree. Once a particular node has been reached, built-in properties can be used to obtain the value of the node, and use it within the script. This tree-based paradigm does away with the problems inherent in SAX's sequential approach, allowing for immediate random access to any node or collection of nodes in the tree. DOM is a W3C Recommendation with ongoing activity.

DOM parsers are available for a variety of different language and platforms, including Perl, PHP, Python, Java , and C/ C++.

The quick lists of the software needed to start Xerces includes:

- The Java Development Kit (JDK), available from the Sun Microsystems Web site <http://java.sun.com> .
- The Apache Web server, available from the Apache Software Foundation's Web site <http://httpd.apache.org>.
- The Tomcat Application Server, available from the Apache Software Foundation's Web site <http://httpd.apache.org>.
- The Xerces parser, available from the Apache XML Project's Web site <http://xml.apache.org>.
- The mod_jk extension for Apache-Tomcat communication, available from the Jakarta Project's Web site <http://httpd.apache.org>.

Installation instructions for all these packages are available in their respective source archives. further installation information is available at http://www.devshed.com/Server_Side/Java/JSPDev , or at the Tomcat User Guide at <http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-ug.html/>. [Icarus 2002]

4. HTML-kit

HTML-Kit is an augmented text editor, and an integrated development (IDE) environment designed to help HTML, XHTML, XML, and script authors to edit, format, lookup help, validate, preview and publish web pages. It maintains full control over multiple file types including HTML, XHTML, XML, CSS, XSL, JavaScript, VBScript, ASP, PHP, JSP, Perl, Python, Ruby, Java, VB, C/C++, C#, Delphi / Pascal, Lisp, SQL, etc.

In addition, HTML-Kit includes internal, external, server-side and live preview modes; File Transfer Protocol (ftp) Workspace for uploading, downloading and online editing of files; and the ability to use hundreds of optional add-ins through its open plugins interface. The latest HTML-Kit release adds the ability to run batch actions such as global search and replace on multiple files and web sites, Internet Explorer and Mozilla/ Netscape side-by-side previewing, W3C Web Accessibility Initiative (WAI) Content Guidelines checking through HTML Tidy, internal Command Prompt, Time-Tracker, translations, Text to Speech Wizard, Unicode-Pad and more [Cham 02].

Figure IV.4 shows a screen Capture where the main window of HTML-KIT contains three sub-windows tiled vertically from the left to the right: a source code of an html file, a preview of another html file, and an internet web page.

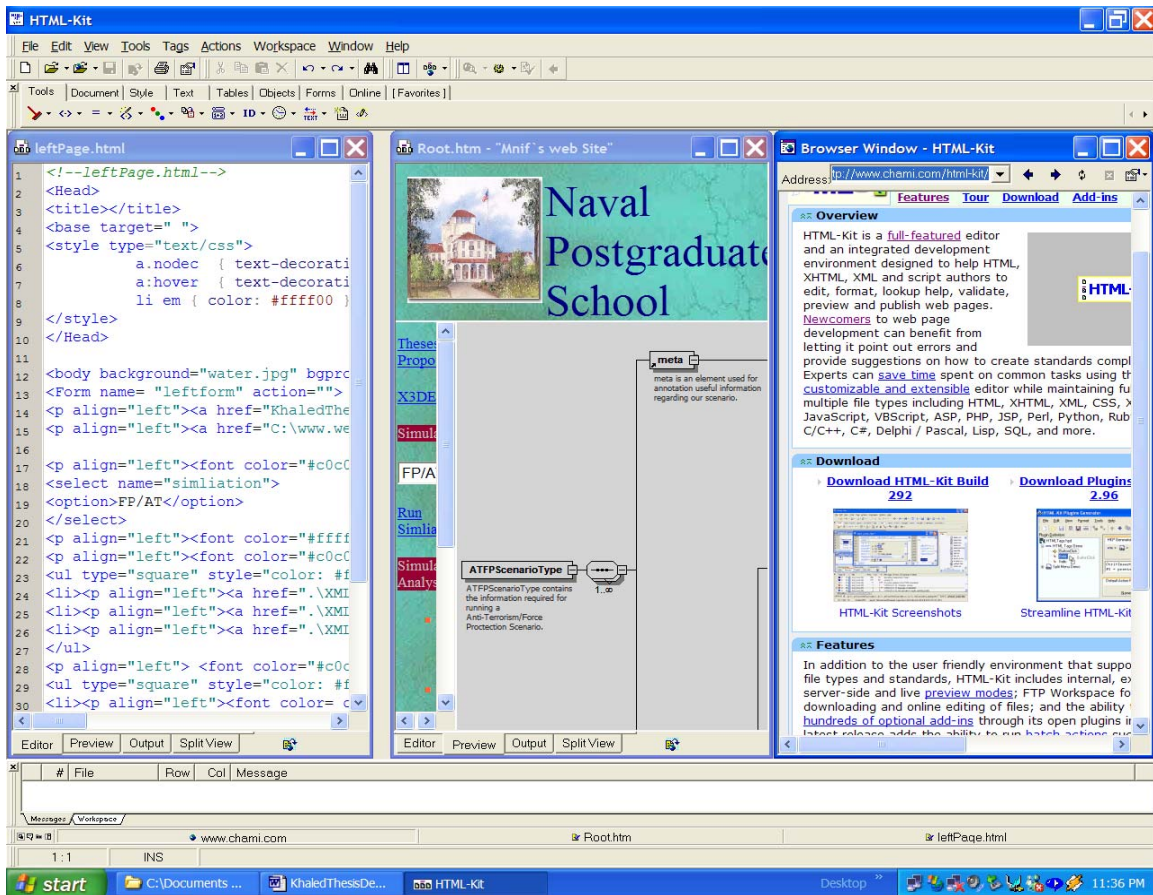


Figure IV.4 Screen Capture of HTML-KIT tool in action-same page w/figure

To learn more and to download HTML-KIT, refer to <http://www.chami.com/html-kit/>

5. Java Servlets

Servlets are Java programs that run on a Web server, acting as a middle layer between a request coming from a Web browser or other HTTP client and database or applications on the HTTP server. Java source code is contained within XML tag for compilation (and caching) on a server. A servlet's job is to:

- Read data sent by the user that is usually entered in a form on a Web page, Java applet, or a custom HTTP client program.
- Look up any other information embedded in the HTTP request. This information includes details about browser capabilities, cookies, the host name of the requesting client, and so forth.

- Generate the results. This process may require talking to a database, executing a Remote Method Invocation (RMI), or a Common Object Request Broker Architecture (COBRA) call, invoking a legacy application or computing the response directly.
 - Format the results inside a document. In most cases, this involves embedding the information inside HTML pages.
 - Set the appropriate HTTP response parameters. This means telling the browser what type of document is being returned (e.g. HTML), setting cookies, caching parameters and other such tasks.
 - Send the document back to the client. This document may be sent in text format (HTML), binary format (GIF images), or even in a compressed format like gzip.
- [Hall 2000]

The Web server executes Servlets inside the Java Virtual Machine (JVM). Figure IV.5 shows the relationship Servlet engine and Web server.

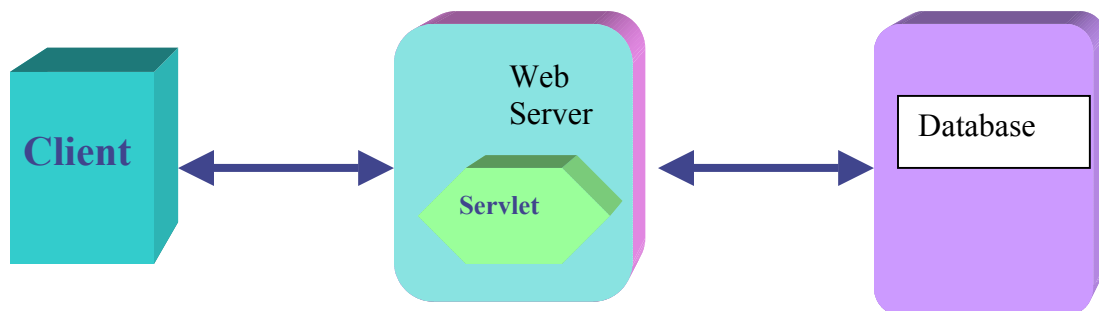


Figure IV.5. Java Servlet Engine and Web Server Relationship

C. ARCHITCTURE OF SYSTEM COMPONENTS

The ATFP scenario results are XML documents. These XML documents are stored in the NPS terra.cs.nps.navy.mil server to compose the database informational needed for displaying and annotating the properties of each entity, tactical data, and statistics of a scenario run.

The user interface is an HTML file that allows a client to request the appropriate data to be displayed. The corresponding XML file is loaded and referenced to a stylesheet that processes and transforms the XML document to an HTML file, then sends it back to the client over HTTP. Figure IV.6 describes the architecture of system components.

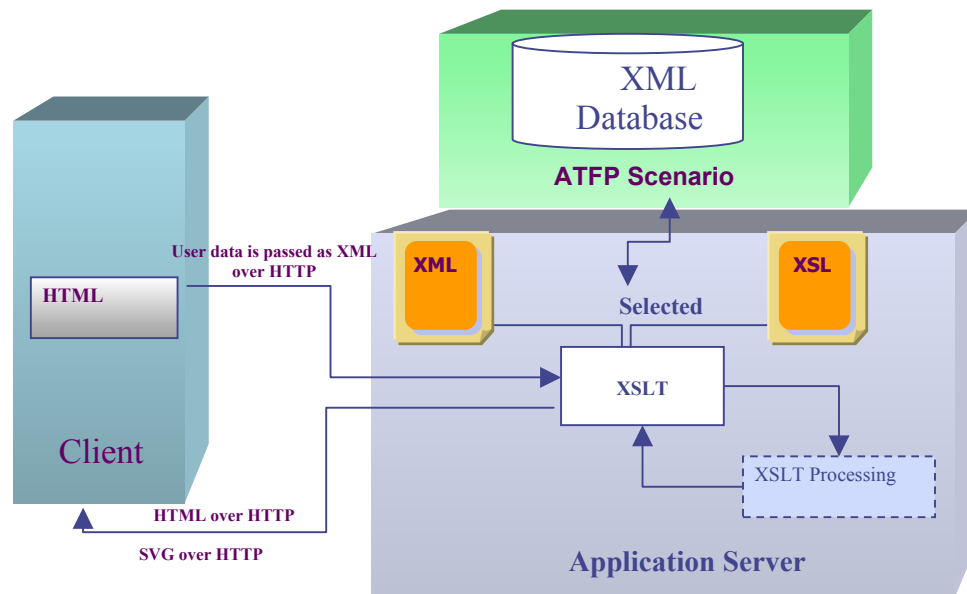


Figure IV.6. Architecture of system components

D. SUMMARY

This chapter describes basic tools used to process and achieve the tasks needed for this research. It then describes the system architecture used for the tactical database system constructed in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

V. TACTICAL SCENARIO DESCRIPTION

A. INTRODUCTION

This chapter describes the components of the Anti-Terrorist/Force Protection (AT/FP) scenario simulating the attack on the USS COLE that occurred in October 2000. LT James Harney has created these scenario components using X3D tool for representing the 3D view of the models.

B. SCENARIO'S COMPONENTS

A geographic locale and agents such as ship, boat, and terrorist are the basic components of the ATFP scenario. All these components are suitable for storage in an XML database.

1. Scenario's Entities

Each scenario agent is considered as an entity in the ATFP scenario. Besides the location component, described in the next section. The basic entities of the scenario are:

- The DDG-51, Arleigh Burke Class Destroyer used in this scenario is the in-point target to be defeated. The primary source of information for the Arleigh Burke model was collected, as mentioned in Chapter III section D, from the web site: <http://www.fas.org/man/dod-101/sys/ship/ddg-51.htm>.

- The standard navy rigid-hull-inflatable boat (RHIB used in this scenario to guard and defend the Arleigh Burke Class Destroyer). Figure V.1 represents a 3D view of the RHIB developed for the ATFP scenario [Harney 03].

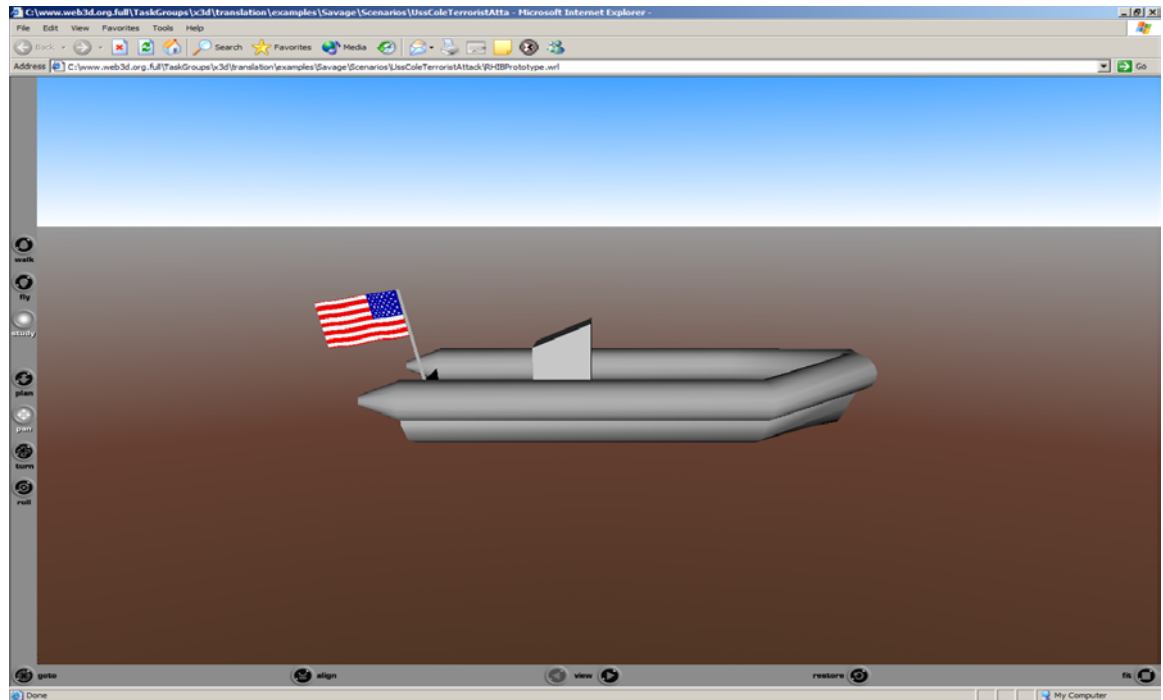


Figure V.1. Screen Snapshot of the RHIB used for ATFP Scenario

[From Harney 2003]

- The Terrorist Boat used to attack the COLE. Figure V.2 shows a 3D view of the Terrorist Boat used to attack the COLE with the Boxman.wrl humanoid used as the ship driver [Harney 2003]

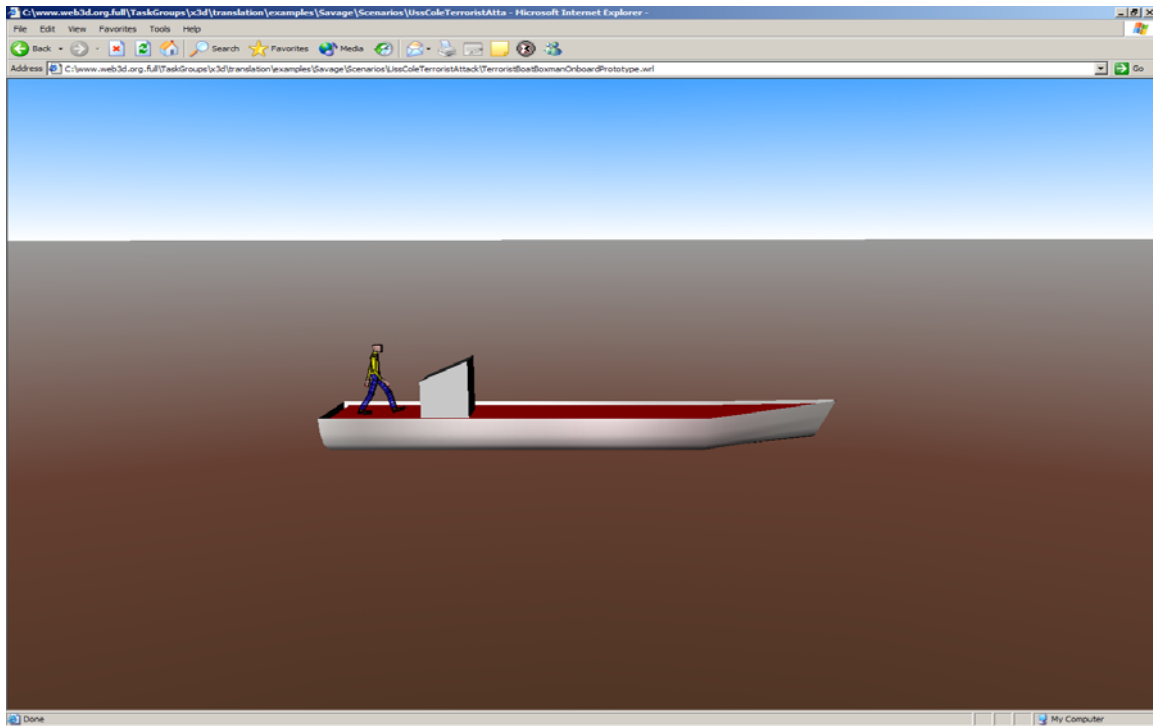


Figure V.2 Screen Snapshot view of the Terrorist Boat used to attack the COLE

[From Harney 2003]

Each entity has its identification and tactical properties. The identification properties of an entity constitute the majority of what the entity model represents. These properties define name, dimensions, speed and other specification of an entity. The tactical properties define all tactical information for an entity such as weapon characteristics, line of sight distance, and defense parameters. Chapter VI includes an XML schema that explains in these entity properties more detail.

2. Location

The geographical location of the ATFP Scenario represents Aden Harbor, Yemen. This model is created by modifying the MatLab script utilized by [BLAIS 01] in the creation of Camp Pendleton, CA). The X3D scene is available online at <http://web.nps.navy.mil/~brutzman/Savage/Scenarios/UssColeTerroristAttack/AdenHarbor.wrl>. This location is defined by its upper left geographic coordinate and its lower right geographic coordinate. Figure V.3 shows a screen capture of Aden Harbor, Yemen X3D scene utilized for reconstruction of the terrorist attack on the USS COLE (DDG 67).

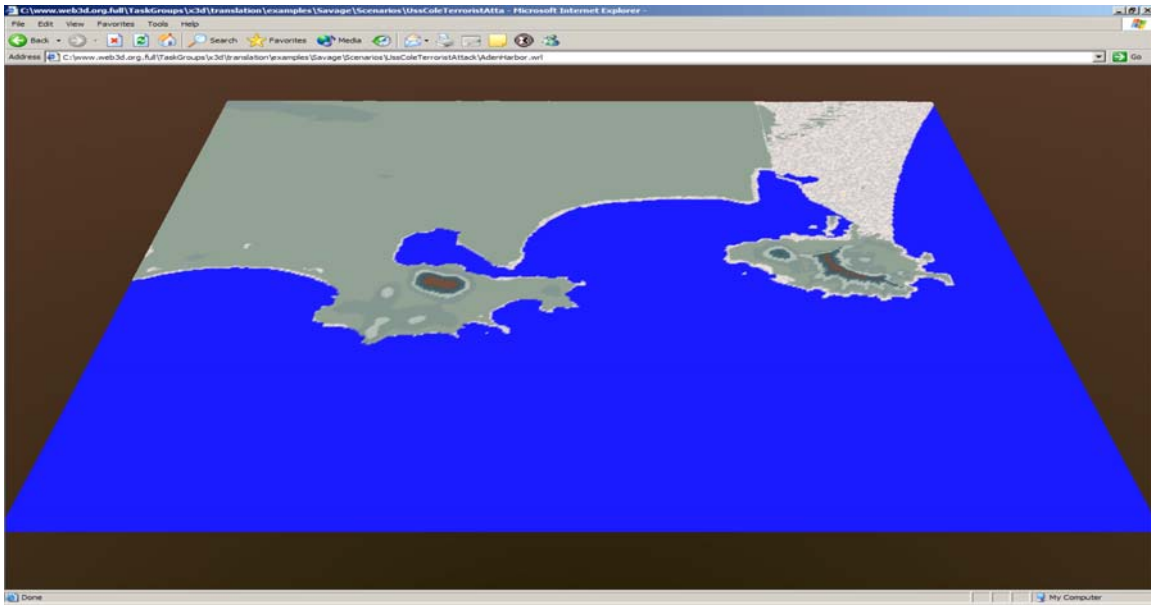


Figure V.3. Screen capture of Aden Harbor, Yemen X3D scene
[From Harney 2003]

C. REQUIREMENTS FOR SCENARIO CREATION

Once the ATPF application is loaded and installed, the end-user has to perform the basic following steps to create a scenario:

- Choose how to configure the scenario. There are three separate options:
 - Use the wizard to configure a Scenario
 - Configure a new Scenario manually
 - Open an existing Scenario
- Select a harbor from the available listing of ports in the menu selection, corresponding to the location of the scenario.
- Configure the frequency of background shipping for the given port or area through a simple radio button assignment and selection.
- Select a high value unit (HVV) to be detected from the available listing of ships in the menu selection. “Once the user has selected a high value unit, specific information on the platform is displayed in the fashion of a simple web-browser from cached unclassified information from the Federation of

American Scientists (FAS.org) web site. If the client machine is on an active network, then this information is displayed live and hyperlinks are selectable for further information searching to aid in any training needs the user may have.” [Harney 2003]

- Configure the defensive setup for the current scenario:
 - Placement of a defending picket boat
 - Configuration of tactical range parameters for identification, intercept, and lethal engagement of approaching craft
 - Select agent control or user control in addition to setting picket boat model parameters for use in the scenario
- Configure the terrorist boat threat to run against the already setup defensive arrangement.

Once all those steps are performed, the scenario is fully configured and the user will have a 2D view on screen as depicted in Figure V.4 [Harney 2003]. Thus, at this point the scenario can be run visually or multiple times for statistics, depending on the user’s needs.

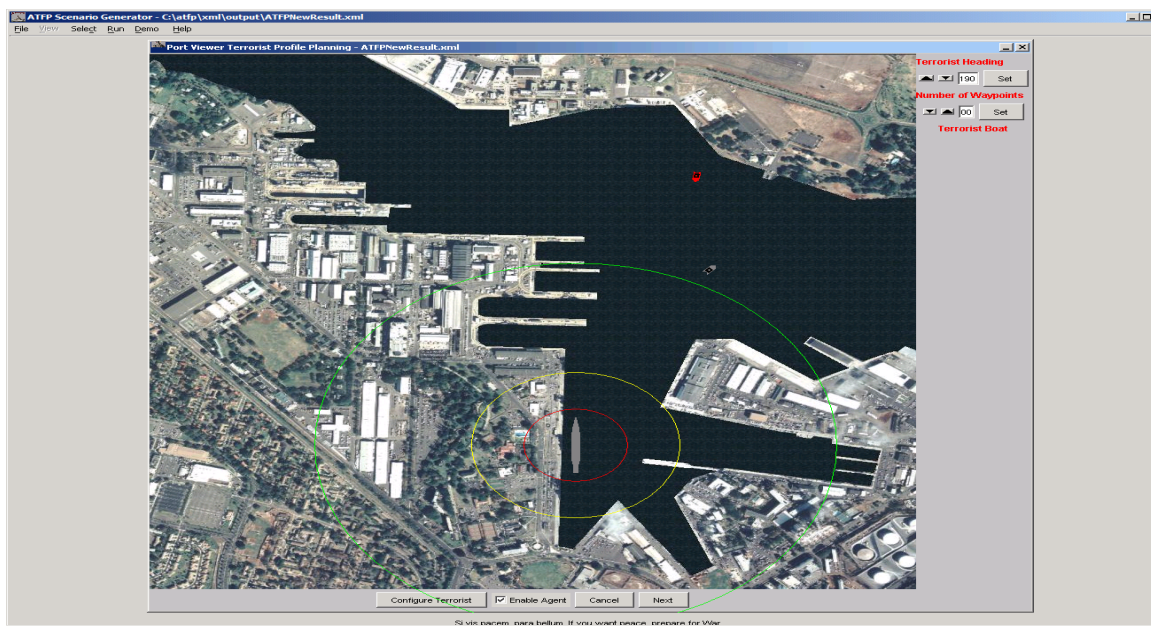


Figure V.4. Example of the terrorist boat attack profile setup [From Harney 2003]

D. REQUIREMENTS FOR STORING SCENARIO RESULTS

After choosing the 3D View option to run the scenario with the setup configuration, the application creates the 3D scenario dynamically and saves the scenario result with the default filename ATFPNewResult.xml if the user has not separately saved the scenario.

The XML schema, defined in Chapter VI table VI.1, is developed for structuring tactical data of AT/FP scenario and validating the ATFPNewResult.xml instance of the scenario result. The XML stylesheets, (Chapter VI tables VI.2 and VI.3), allow the user to display properties and tactical data of the scenario entities and statistics of multiple scenario runs.

E. SUMMARY

The first part of this chapter describes the basic components of the ATFP scenario accompanied by 3D views. The DDG-51, Arleigh Burke Class Destroyer, the standard navy rigid-hull-inflatable-boat (RHIB), a terrorist Boat, and Yemen's Aden Harbor are example of the scenario's components. Each of these components is suitable for storage and retrieval via an XML database. The second part lists the basic steps for creating scenario and storing results. [Harney 2003] describe these steps with more details.

VI. XML SCHEMA FOR ANTI-TERRORIST FORCE PROTECTION SCENARIO REPRESENTATION

A. INTRODUCTION

This chapter defines the XML Schema for AT/FP scenario representation describing the structure of AT/FP XML document, and defines XSLT stylesheets used for transmuting instances of XML documents.

B. XML SCHEMA FOR SCENARIO REPRESENTATION

1. Friendly and Hostile Forces, Location, and Measures Of Effectiveness (MOEs)

The AT/FP XML Schema specifies the type of textual data within attributes and elements using simple type declarations such as positive number, float, and string. These types provide rigid control of XML document elements and attributes that are intended to represent our database. Basically, this Schema describes properties and tactical data for friendly and hostile forces (ships, planes, people, weapons, etc.), tactical data for the scenario location (geographical positions). It also includes measures of effectiveness (MOEs) per scenario run (and per multiple scenario runs) such as overall mean, variance, and standard deviation. Figure VI.1 presents the basic Schema design view of ATFP Scenario generated by XML Spy. Appendix A contains the entire ATFP XML Schema.

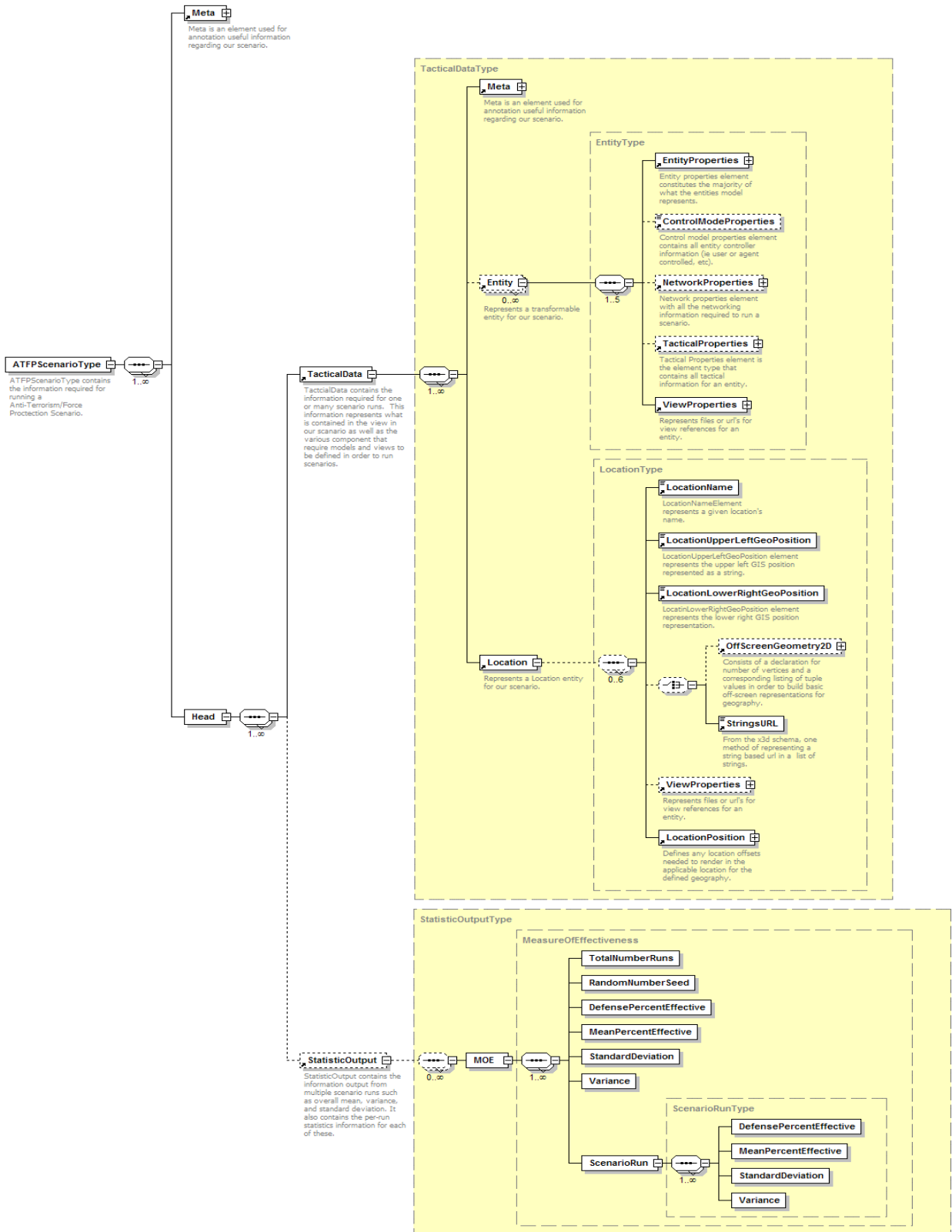


Figure VI.1. Schema design view of ATFP

C. XSLT STYLESHEETS FOR ANTI-TERRORIST SCENARIO

Two W3C Recommendations, XSLT (the Extensible Stylesheet Language Transformations) and XPath (the XML Path Language), provide a powerful implementation of a tree-oriented transformation language for transmuting instances of XML using one vocabulary into either simple text, presentation in HTML, or XML instances using any other vocabulary imaginable. For this purpose, the XSLT language, which itself uses XPath, is used to specify how an implementation of an XSLT processor can create our desired output from a given marked-up input.

1. XSLT Stylesheet for Entities Properties and Tactical Data

Appendix B contains a summary of the pattern-match rules found in the XSLT stylesheet that produces HTML output from an AT/FP document. Chapter VII Section B defines an instance of XML document to which XSLT stylesheet is applied.

2. XSLT Stylesheet For Measure Of Effectiveness

Appendix C presents the XSLT stylesheet, which transform and display Statistics output values after multiple runs of ATFP scenario.

D. SUMMARY

Chapter VI defines the database structure of the ATFP scenario via its XML Schema and gives examples of XSLT stylesheets for transforming and displaying information from XML instances.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. EXPERIMENTAL AND SIMULATION RESULTS

A. INTRODUCTION

This chapter shows an XML document as an instance of our XML database, which is an output of the AT/FP scenario. Then it describes simulation results via screen Captures taken from a client running the application existing in terra server.

B. AT/FP SCENARIO INSTANCE

Appendix D contains an instance of XML document, which is a result of AT/FP scenario runs. This instance is stored in our database and served to the end user as HTML files after applying the stylesheets described in Chapter VI.

C. SCENARIO SCREEN CAPTURES EXPLAINED

1. User Interface

The user interface is an HTML page composed of three frames. The top frame displays NPS logo, and flags for California State and the USA. The left frame contains different options allowing the user to download the ATFP Scenario, run the scenario, display the results, and perform other options. The main frame as shown contains the design view of ATFP database structure as a background. Figure VII.1 is a screen Capture of the user interface loaded from terra.cs.nps.navy.mil server.

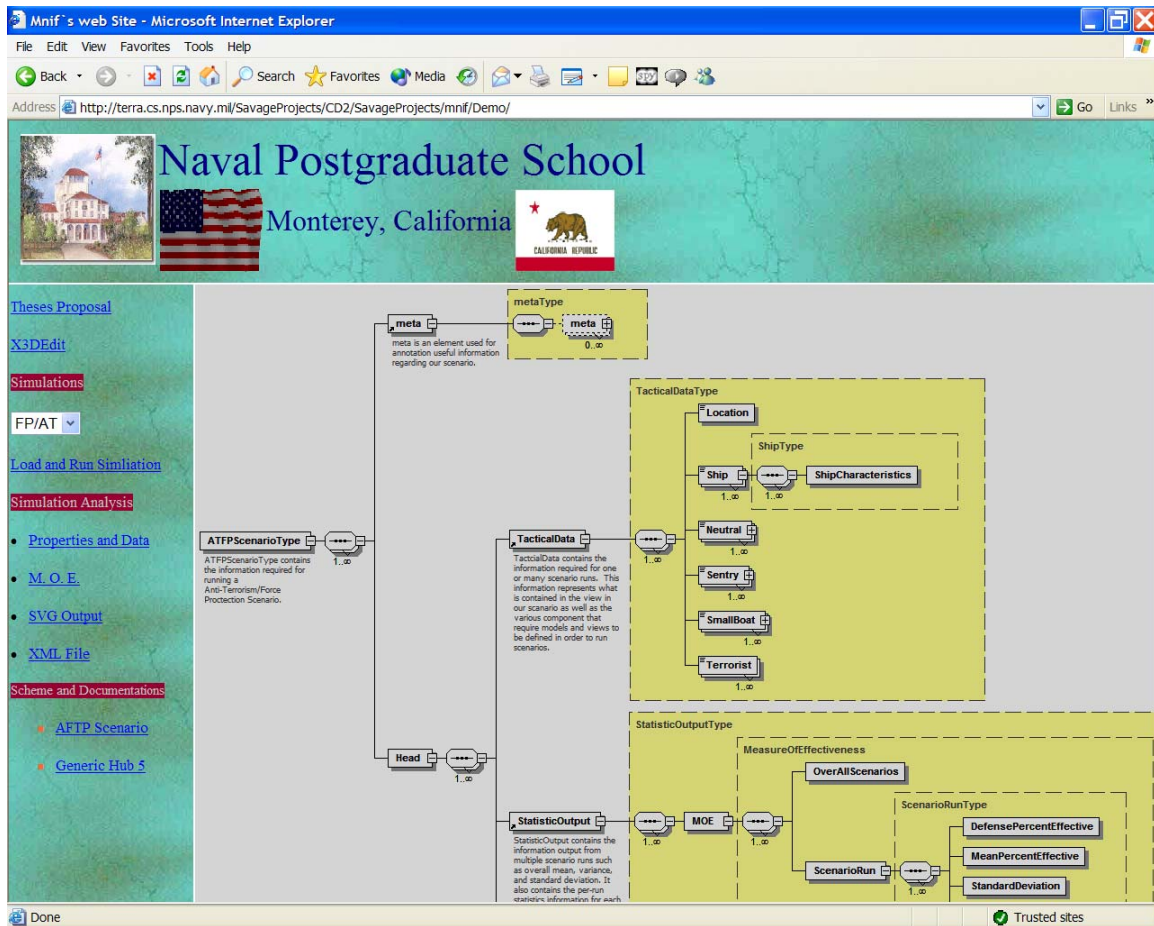


Figure VII.1. Screen Capture of the user interface

2. Simulation Results

a. Scenario Properties and tactical data

After running the ATFTP application, the user can choose “properties and data” option. On the server side, a corresponding XSLT stylesheet is selected to process an XML document, which represents the instance of AT/FP scenario result. The XSLT stylesheet produces an HTML file, which is sent back to the end-user via HTTP displaying the result requested. Figure VII.2 represents a screen Capture showing the result after selecting the “properties and data” option.

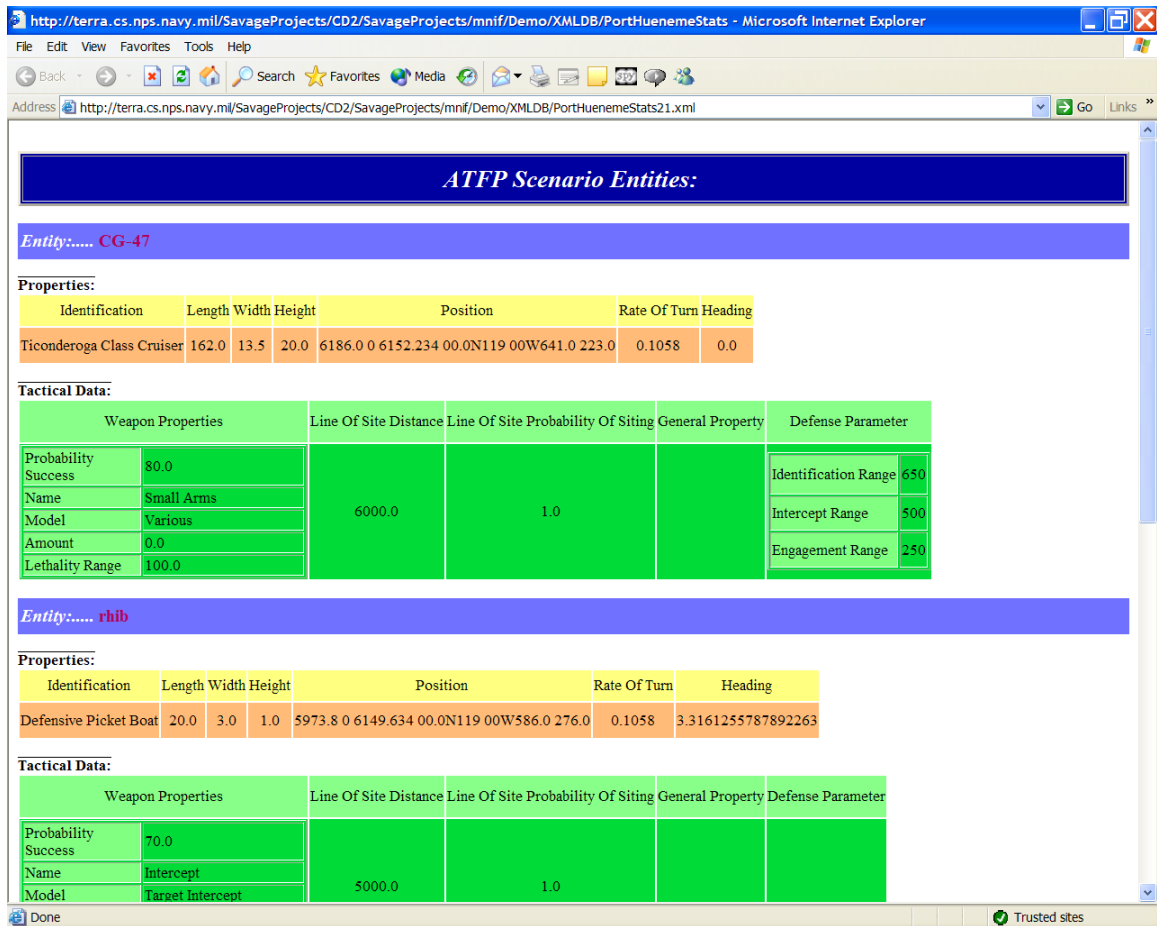


Figure VII.2. Screen Capture displaying the properties and tactical data of an XML instance recording an AT/FP scenario run

b. Scenario Statistic Output

To display the Measure Of Effectiveness (MOE) of the AT/FP scenario runs, another XSLT styleseet, different from the one used to display properties and tactical data, is applied to display the Scenario Statistic output.

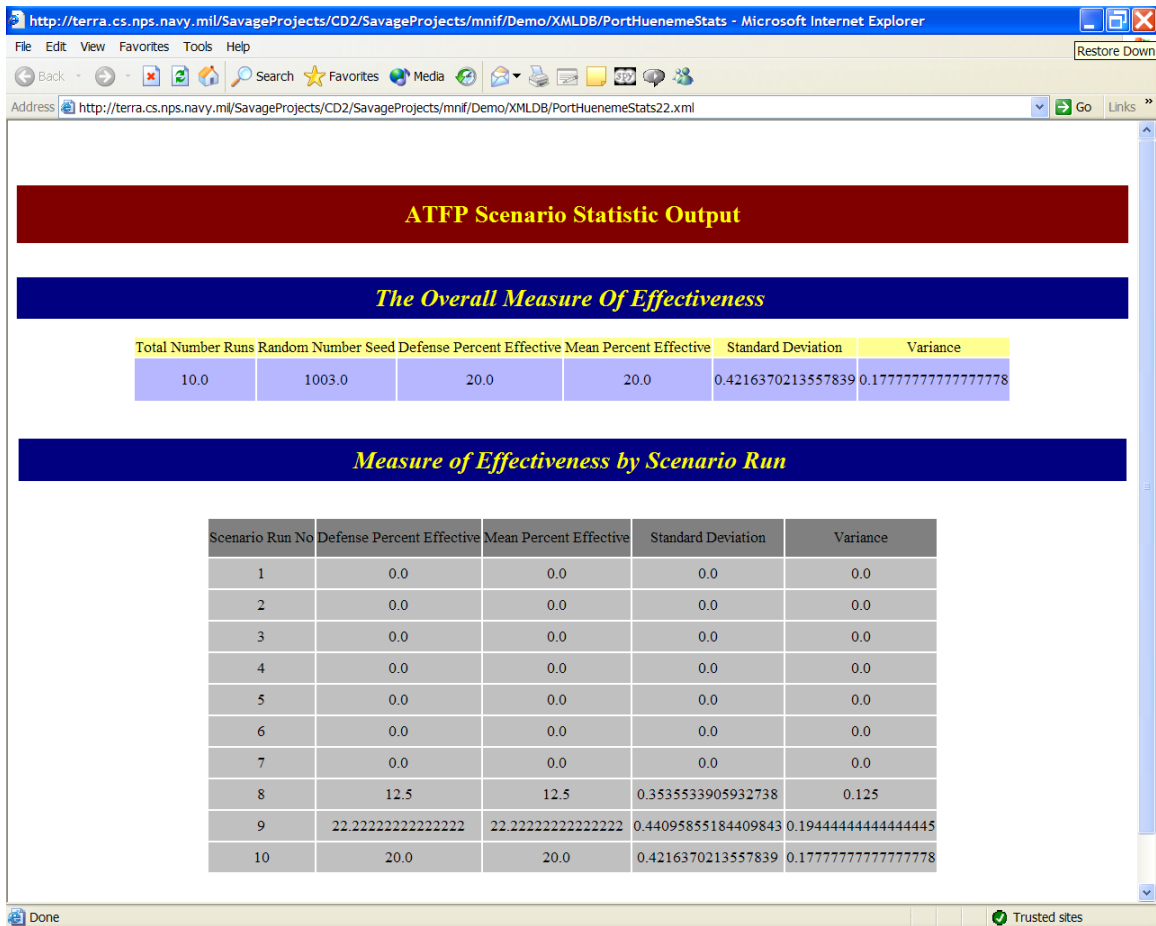


Figure VII.2. Screen Capture displaying the measure of effectiveness (MOE) of the AT/FP scenario runs.

D. SUMMARY

This chapter describes how XSLT stylesheets are applied to an XML instance of the ATFP scenario results. These XSLT stylesheets transform requested data into HTML formats to be sent via HTTP and displayed in the client side.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Many resources provide a database of military models with tactical information. Such tactical data are useful for defining and populating databases. Therefore, a common information-exchange data model can be provided and constantly updated. XML technologies provide this capability well by using either open source or commercial tools.

This research addressed the problem of storing, serving, and annotating Tactical Scenario data for X3D using HTTP and XML technologies for choosing XML for this research are diverse:

- The power of XML is the universal support it is gaining. In this case, the noise is the substance.
- XML opens the door for a new generation of databases. Relational databases will not go away, but a new kind of database will take root; namely, object databases, ones that can manage heterogeneous, non-tabular information. In other words, XML is more general than the relational form. Relational databases will only be able to manage a subset of XML.
- XML works well with all database systems.
- XML is well suited for military messages.
- NATO Battlespace Generic Hub can add semantic consistency tactical conclusions.

B. RECOMMENDATIONS FOR FUTURE WORK

This research represents a prototype implementation of ATFP scenario. It is important to continue to work on an XML schema and XSL stylesheet that automatically generates a 3D visualization. This work can be extended for the US Army using different X3D models with different entities (tanks, vehicles, etc.) and locations.

The Generic Hub could be used to extend ATFP scenario. The proposed scenarios expressed in XML can be mapped to concepts that are contained in the Generic Hub. XSLT transformations use this document to feed scenario information to the ATFP application, and to extract information to be represented in the GH XML instance. Some nodes are used only for input, and some nodes are used only for output. Some functions of the program, like the results of the simulations, are not expressed in the Generic Hub.

The most important recommendation for future work is combining XML technologies and a native XML database such as Xindice, which is an open source tool, to create store, edit, and query the result of multiple simulations.

APPENDIX A. AT/FP XML SCHEMA

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by James Harney (US Naval Postgraduate School) -->
<!--W3C Schema generated by XML Spy v4.3 (http://www.xmlspy.com)-->
<!--Authors: James Harney, Khaled Mnif-->
<!--Created: 2 August 2002-->
<!--Revised: 08 October 2002-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:element name="ATFPScenarioType">
    <xsd:annotation>
      <xsd:documentation>ATFPScenarioType contains the information required for running a Anti-Terrorism/Force
Protection Scenario. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element ref="Meta"/>
        <xsd:element name="Head">
          <xsd:complexType>
            <xsd:sequence maxOccurs="unbounded">
              <xsd:element ref="TacticalData"/>
              <xsd:element ref="StatisticOutput" minOccurs="0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="doubleList">
    <xsd:annotation>
      <xsd:appinfo/>
      <xsd:documentation>doubleList is an internal type for creating lists in the schema, not for author use,
that may get refactored away in a future version of the X3D schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:list itemType="xsd:double"/>
  </xsd:simpleType>
  <xsd:simpleType name="floatList">
    <xsd:annotation>
```

```

        <xsd:appinfo/>
        <xsd:documentation>floatList is an internal type for creating lists in the schema, not for author use, that
may get refactored away in a future version of the X3D schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:list itemType="xsd:float"/>
</xsd:simpleType>

<xsd:simpleType name="integerListType">
    <xsd:annotation>
        <xsd:appinfo/>
        <xsd:documentation>integerList is an internal type for creating lists in the schema, not for author use,
that may get refactored away in a future version of the X3D schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:list itemType="xsd:integer"/>
</xsd:simpleType>
<xsd:simpleType name="Vector3Double">
    <xsd:annotation>
        <xsd:appinfo/>
        <xsd:documentation source="http://www.geovrml.org/1.0/doc/concepts.html">
            Vector3Double is a 3-tuple triplet of Double values.
            See GeoVRML 1.0 Recommended Practice, Section 2.3, Limitations Of Single-Precision.
            Hint: Vector3Double can be used to specify a georeferenced 3D coordinate.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="doubleList">
            <xsd:maxLength value="3"/>
            <xsd:pattern value="zero or three doubles"/>
        </xsd:restriction>
    </xsd:simpleType>
<xsd:simpleType name="Vector3Float">
    <xsd:annotation>
        <xsd:appinfo/>
        <xsd:documentation
source="http://www.web3D.org/technicalinfo/specifications/vrml97/part1/fieldsRef.html#SFVec3f">
            Vector3Float is a 3-tuple triplet of Float values.
            Vector3Float is equivalent to type SFVec3f in the VRML 97 Specification.
            Hint: Vector3Float can be used to specify a 3D coordinate or a 3D scale value.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="floatList">
            <xsd:maxLength value="3"/>
        </xsd:restriction>
    </xsd:simpleType>

```

```

<xsd:attribute name="description" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>General description of the scenario.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="name" type="xsd:string" use="required">
  <xsd:annotation>
    <xsd:documentation>name attribute denotes the string based name for the scenario.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:element name="TacticalData" type="TacticalDataType">
  <xsd:annotation>
    <xsd:documentation>TacticalData contains the information required for one or many scenario runs. This
information represents what is contained in the view in our scenario as well as the various component that require
models and views to be defined in order to run scenarios.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="StatisticOutput" type="StatisticOutputType">
  <xsd:annotation>
    <xsd:documentation>StatisticOutput contains the information output from multiple scenario runs such as
overall mean, variance, and standard deviation. It also contains the per-run statistics information for each of
these.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="DefensePercentEffectiveType">
  <xsd:annotation>
    <xsd:documentation>DefensePercentEffectiveType represents the overall effectiveness of a given defense
where 0.0 represents total success of a terrorist attack and 1.0 represents a full-proof
defense.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="value" type="xsd:float" use="required"/>
</xsd:complexType>
<xsd:complexType name="DefenseParameterType">
  <xsd:annotation>
    <xsd:documentation> Defense Parameter type can be used to represent different tactical parameters in our
scenario. Note not currently using.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence maxOccurs="3">
    <xsd:element ref="IdentificationRange"/>
    <xsd:element ref="InterceptRange"/>
    <xsd:element ref="EngagementRange"/>
  </xsd:sequence>

```

```

</xsd:complexType>
<xsd:complexType name="MeasureOfEffectiveness">
  <xsd:annotation>
    <xsd:documentation>MeasureOfEffectiveness denotes the overall measure that our scenario is being based
upon.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="TotalNumberRuns" type="TotalNumberRunsType"/>
    <xsd:element name="RandomNumberSeed" type="RandomNumberSeedType"/>
    <xsd:element name="DefensePercentEffective" type="DefensePercentEffectiveType"/>
    <xsd:element name="MeanPercentEffective" type="MeanPercentEffectiveType"/>
    <xsd:element name="StandardDeviation" type="StandardDeviationType"/>
    <xsd:element name="Variance" type="VarianceType"/>
    <xsd:element name="ScenarioRun" type="ScenarioRunType"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:complexType name="MeanPercentEffectiveType">
  <xsd:annotation>
    <xsd:documentation>MeanPercentEffectiveType denotes the overall mean percent effective.</xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="value" type="xsd:float" use="required"/>
</xsd:complexType>
<xsd:complexType name="MetaType">
  <xsd:annotation>
    <xsd:documentation>MetaType is a complex type used for annotating useful information for various components
of our scenario</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Meta" type="MetaType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="content" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="Location"/>
        <xsd:enumeration value="Neutral"/>
        <xsd:enumeration value="Sentry"/>
        <xsd:enumeration value="Ship"/>
        <xsd:enumeration value="Simulation"/>
        <xsd:enumeration value="SmallBoat"/>
        <xsd:enumeration value="Terrorist"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
<xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>
<xsd:element name="Meta" type="MetaType">
    <xsd:annotation>
        <xsd:documentation>Meta is an element used for annotation useful information regarding our
scenario.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="RandomNumberSeedType">
    <xsd:annotation>
        <xsd:documentation>RandomNumberSeedType denotes the random number seed used for this scenario
run.</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="value" type="xsd:double" use="required"/>
</xsd:complexType>
<xsd:complexType name="ScenarioRunType">
    <xsd:annotation>
        <xsd:documentation>ScenarioRunType denotes the overall scenario run type.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="DefensePercentEffective" type="DefensePercentEffectiveType"/>
        <xsd:element name="MeanPercentEffective" type="MeanPercentEffectiveType"/>
        <xsd:element name="StandardDeviation" type="StandardDeviationType"/>
        <xsd:element name="Variance" type="VarianceType"/>
    </xsd:sequence>
    <xsd:attribute name="value" type="xsd:double" use="required"/>
</xsd:complexType>
<xsd:complexType name="StandardDeviationType">
    <xsd:annotation>
        <xsd:documentation>StandardDeviationType denotes the standard deviation for a given set or single scenario
runs.</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="value" type="xsd:float" use="required"/>
</xsd:complexType>
<xsd:complexType name="StatisticOutputType">
    <xsd:annotation>
        <xsd:documentation>StatisticOutputType represents a collection of statistics for a given
scenario.</xsd:documentation>
    </xsd:annotation>

```



```

        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
            <xsd:element name="MOE" type="MeasureOfEffectiveness"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TacticalDataType">
        <xsd:annotation>
            <xsd:documentation>TacticalDataType represents given tactical data for a given
scenario.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element ref="Meta"/>
            <xsd:element ref="Entity" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="Location"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="TotalNumberRunsType">
        <xsd:annotation>
            <xsd:documentation>TotalNumberRunsType represents the total number of runs when running multiple scenarios
off-screen for multiple runs.</xsd:documentation>
        </xsd:annotation>
        <xsd:attribute name="value" type="xsd:double" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="VarianceType">
        <xsd:annotation>
            <xsd:documentation>VarianceType represents the variance value for a given scenario run or multiple
runs.</xsd:documentation>
        </xsd:annotation>
        <xsd:attribute name="value" type="xsd:float" use="required"/>
    </xsd:complexType>
    <xsd:element name="DefenseTacticalParameters">
        <xsd:annotation>
            <xsd:documentation>Defense Tactical Parameters can represent different tactical parameters in our
scenario.</xsd:documentation>
        </xsd:annotation>
        <xsd:complexType>
            <xsd:attribute name="description" type="floatList"/>
        </xsd:complexType>
    </xsd:element>
    <!--Start New Entries 08 Oct 2002-->
    <xsd:complexType name="EntityDataType">
        <xsd:annotation>
            <xsd:documentation>Contains entity property data with unspecified properties being allowed in the

```

```

GeneralProperty type.</xsd:documentation>
</xsd:annotation>
<xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="Name"/>
  <xsd:element ref="Identification"/>
  <xsd:element ref="Length" minOccurs="0"/>
  <xsd:element ref="Width" minOccurs="0"/>
  <xsd:element ref="Height" minOccurs="0"/>
  <xsd:element ref="MinimumSpeed" minOccurs="0"/>
  <xsd:element ref="MaximumSpeed" minOccurs="0"/>
  <xsd:element ref="GeneralProperty" minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element ref="EntityPosition"/>
  <xsd:element ref="EntityRateOfTurn" minOccurs="0"/>
  <xsd:element ref="EntityHeading"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="GeneralProperty" type="GeneralPropertyType">
  <xsd:annotation>
    <xsd:documentation>General element that allows miscellaneous properties to be entered for an
entity.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="GeneralPropertyType">
  <xsd:annotation>
    <xsd:documentation>General Property type allows miscellaneous properties to be entered for an entity
without having to modify our schema.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="2" maxOccurs="2">
    <xsd:element ref="Name"/>
    <xsd:element ref="Value"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:attribute name="value" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>value is a generalized string attribute usefult within our schema.</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:simpleType name="LengthType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the length for an entity if they have one.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>

```

```

</xsd:simpleType>
<xsd:simpleType name="WidthType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the width for an entity if they have one.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="HeightType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the height for an entity if they have one.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="MaximumSpeedType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the maximum speed for an entity if they have
one.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="MinimumSpeedType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the minimum speed for an entity if they have
one.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="NameType">
  <xsd:annotation>
    <xsd:documentation>Type that represents the name for an entity.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="IdentificationType">
  <xsd:annotation>
    <xsd:documentation>Type that represents an entities identification type separate from its
name.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="Length" type="LengthType">
  <xsd:annotation>

```

```

        <xsd:documentation>Element representing an entities length.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Width" type="WidthType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities width.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Height" type="HeightType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities height.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="MaximumSpeed" type="MaximumSpeedType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities Maximum speed.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="MinimumSpeed" type="MinimumSpeedType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities Minimum speed.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Name" type="NameType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities name.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Identification" type="IdentificationType">
    <xsd:annotation>
        <xsd:documentation>Element representing an entities Identification separate and different from the entities
name.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="EntityProperties" type="EntityDataType">
    <xsd:annotation>
        <xsd:documentation>Entity properties element constitutes the majority of what the entities model
represents.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<!--Network Property simple, complex types and element definitions-->
<xsd:simpleType name="ProtocolType">

```

```

    <xsd:annotation>
      <xsd:documentation>Type to represent a network prototcol's identification.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="IPAddressType">
    <xsd:annotation>
      <xsd:documentation>Type to represent a standard IP address.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:simpleType name="EntityIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Type to represent an entity id in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:simpleType name="ApplicationIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Type to represent an entity application id in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:simpleType name="SiteIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Type to represent an entity site id in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:simpleType name="EntityKindType">
    <xsd:annotation>
      <xsd:documentation>Type to represent EntityKind in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:simpleType name="EntityDomainType">
    <xsd:annotation>
      <xsd:documentation>Type to represent Entity Domain in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:simpleType name="EntityCountryType">

```

```

    <xsd:annotation>
      <xsd:documentation>Type to represent Entity Country in DIS.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:integer"/>
  </xsd:simpleType>
  <xsd:element name="NetworkProtocolPrimary" type="ProtocolType">
    <xsd:annotation>
      <xsd:documentation>Primary network protocol element.</xsd:documentation>
    </xsd:annotation>

  <xsd:element name="IPAddress" type="IPAddressType">
    <xsd:annotation>
      <xsd:documentation>Element that represents an IPAddress.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="EntityIdentification" type="EntityIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Element that represents a DIS Entity ID.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="ApplicationIdentification" type="ApplicationIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Element that represents a DIS Application ID.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="SiteIdentification" type="SiteIdentificationType">
    <xsd:annotation>
      <xsd:documentation>Element that represents a DIS Site ID</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="EntityKind" type="EntityKindType">
    <xsd:annotation>
      <xsd:documentation>Element that represents a DIS Entity Kind enumeration.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="EntityDomain" type="EntityDomainType">
    <xsd:annotation>
      <xsd:documentation>Element that represents a DIS EntityDomain enumeration.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="EntityCountry" type="EntityCountryType">
    <xsd:annotation>

```

```

    <xsd:documentation>Element that represents a DIS Entity Country enumeration.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="NetworkPropertyType">
  <xsd:annotation>
    <xsd:documentation>Type that represents all Network properties in one place.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="NetworkProtocolPrimary"/>
    <xsd:element ref="IPAddress"/>
    <xsd:element ref="GeneralProperty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="DISInformation" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="NetworkProperties" type="NetworkPropertyType">
  <xsd:annotation>
    <xsd:documentation>Network properties element with all the networking information required to run a
scenario.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!--Weapon Property types and elements-->
<xsd:simpleType name="ProbabilityOfWeaponSuccessType">
  <xsd:annotation>
    <xsd:documentation>Type for representing Probability Of weapon's success.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="WeaponNameType">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a weapon name type.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="WeaponModelType">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a weapon model type.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="WeaponAmountType">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a weapon amount type.</xsd:documentation>

```

```

</xsd:annotation>
<xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="WeaponLethalityRange">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a weapon lethality range type.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:complexType name="WeaponType">
  <xsd:annotation>
    <xsd:documentation>Complex type representing all weapon properties in one place as sub-
components.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="ProbabilityOfWeaponSuccess" minOccurs="0"/>
    <xsd:element name="WeaponName"/>
    <xsd:element name="WeaponModel" minOccurs="0"/>
    <xsd:element name="WeaponAmount" minOccurs="0"/>
    <xsd:element name="WeaponLethalityRange" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="ProbabilityOfWeaponSuccess" type="ProbabilityOfWeaponSuccessType">
  <xsd:annotation>
    <xsd:documentation>Element representing probability of weapon success.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="WeaponName" type="WeaponNameType">
  <xsd:annotation>
    <xsd:documentation>Element representing a weapon's name.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="WeaponModel" type="WeaponModelType">
  <xsd:annotation>
    <xsd:documentation>Element representing a weapon's model.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="WeaponAmount" type="WeaponAmountType">
  <xsd:annotation>
    <xsd:documentation>Element representing amount of a weapon we have.</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```



```

<xsd:element name="WeaponLethalityRange" type="WeaponLethalityRange">
  <xsd:annotation>
    <xsd:documentation>Element representing a weapon's lethality range.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="WeaponProperties" type="WidthType">
  <xsd:annotation>
    <xsd:documentation>Element representing all the weapon's properties.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!--Other types and elements used for TacticalProperties-->
<xsd:simpleType name="LineOfSiteDistanceType">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a line of sight distance cap for our
entity.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="LineOfSiteProbabilityOfSitingType">
  <xsd:annotation>
    <xsd:documentation>Simple type representing a line of sight probability of deterction
threshold.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:complexType name="TacticalPropertyType">
  <xsd:annotation>
    <xsd:documentation>Contains tactical data for an entity such as any weapons and their properties,
etc.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="WeaponProperties" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="LineOfSiteDistance" minOccurs="0"/>
    <xsd:element ref="LineOfSiteProbabilityOfSiting" minOccurs="0"/>
    <xsd:element ref="GeneralProperty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="DefenseParameter"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="LineOfSiteDistance" type="LineOfSiteDistanceType">
  <xsd:annotation>
    <xsd:documentation>Straight line of site distance this entity can normally see independent of any
enviromental factors.</xsd:documentation>

```

```

    </xsd:annotation>
  </xsd:element>
  <xsd:element name="LineOfSiteProbabilityOfSiting" type="LineOfSiteProbabilityOfSitingType">
    <xsd:annotation>
      <xsd:documentation>Probability factor that can be applied to the LineOfSiteDistance property factor for an
entity.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="TacticalProperties" type="TacticalPropertyType">
    <xsd:annotation>
      <xsd:documentation>Tactical Properties element is the element type that contains all tactical information
for an entity.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <!--Controller Properties for entities-->
  <xsd:simpleType name="ControlModeType">
    <xsd:annotation>
      <xsd:documentation>Indicates agent or User control of the entity.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:element name="ControlModeProperties" type="ControlModeType">
    <xsd:annotation>
      <xsd:documentation>Control model properties element contains all entity controller information (ie user or
agent controlled, etc).</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="ControlPropertyType">
    <xsd:annotation>
      <xsd:documentation>ControlPropertyType is the type representing all entity controller
information.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element name="ControlMode"/>
      <xsd:element name="GeneralProperty" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <!--Location Information Types and Element definitions-->
  <xsd:simpleType name="LocationUpperLeftGeoPositionType">
    <xsd:annotation>
      <xsd:documentation>LocationUppLeftGeoPositionType represen ts the upper left geo position anchor for
geometry.</xsd:documentation>

```

```

        </xsd:annotation>
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="LocationLowerRightGeoPositionType">
        <xsd:annotation>
            <xsd:documentation>LocationLowerRightGeoPositionType represents the lower right geo position anchor for
geometry.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:element name="LocationUpperLeftGeoPosition" type="LocationUpperLeftGeoPositionType">
        <xsd:annotation>
            <xsd:documentation>LocationUpperLeftGeoPosition element represents the upper left GIS position represented
as a string.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="LocationLowerRightGeoPosition" type="LocationLowerRightGeoPositionType">
        <xsd:annotation>
            <xsd:documentation>LocatinLowerRightGeoPosition element represents the lower right GIS position
representation.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:simpleType name="LocationNameType">
        <xsd:annotation>
            <xsd:documentation>LocationNameType represents the string based name for a given
location.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:element name="LocationName" type="LocationNameType">
        <xsd:annotation>
            <xsd:documentation>LocationNameElement represents a given location's name.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="LocationPropertyType">
        <xsd:annotation>
            <xsd:documentation>LocationPropertyType represents basic location properties for the Location
entity.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence maxOccurs="unbounded">
            <xsd:element ref="LocationName"/>
            <xsd:element ref="LocationUpperLeftGeoPosition"/>

```

```

        <xsd:element ref="LocationLowerRightGeoPosition"/>
    </xsd:sequence>
</xsd:complexType>
<!--Primary Data types and elements to be used, ie Entity, Location, etc.-->
<xsd:element name="Entity" type="EntityType">
    <xsd:annotation>
        <xsd:documentation>Represents a transformable entity for our scenario.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Location" type="LocationType">
    <xsd:annotation>
        <xsd:documentation>Represents a Location entity for our scenario.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="LocationType">
    <xsd:annotation>
        <xsd:documentation>Defines the Location entity. Consists of a name, upper left, and lower right posits.
Also offscreen representations of the geography.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence minOccurs="0" maxOccurs="6">
        <xsd:element ref="LocationName"/>
        <xsd:element ref="LocationUpperLeftGeoPosition"/>
        <xsd:element ref="LocationLowerRightGeoPosition"/>
        <xsd:choice minOccurs="0">
            <xsd:element ref="OffScreenGeometry2D" minOccurs="0"/>
            <xsd:element ref="StringsURL"/>
        </xsd:choice>
        <xsd:element ref="ViewProperties" minOccurs="0"/>
        <xsd:element ref="LocationPosition"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EntityType">
    <xsd:annotation>
        <xsd:documentation>Defines the entity type that can represent ships, small boats, etc.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence maxOccurs="5">
        <xsd:element ref="EntityProperties"/>
        <xsd:element ref="ControlModeProperties" minOccurs="0"/>
        <xsd:element ref="NetworkProperties" minOccurs="0"/>
        <xsd:element ref="TacticalProperties" minOccurs="0"/>
        <xsd:element ref="ViewProperties"/>
    </xsd:sequence>

```

```

</xsd:complexType>
<xsd:simpleType name="VerticeNumberType">
  <xsd:annotation>
    <xsd:documentation>Represents the number of vertices that are in our off-screen representation for
polygons</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:integer"/>
</xsd:simpleType>
<xsd:complexType name="OffScreenGeometry2DType">
  <xsd:annotation>
    <xsd:documentation>Represents off-screen geometry in 2D. Used for representation of land boundaries with
water and so forth.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0" maxOccurs="2">
    <xsd:element ref="VerticeNumber"/>
    <xsd:element ref="IntegerList"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="VerticeNumber" type="VerticeNumberType">
  <xsd:annotation>
    <xsd:documentation>Number of vertices for the off-screen geometry representation.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="IntegerList" type="integerListType">
  <xsd:annotation>
    <xsd:documentation>From the X3D Schema, this allows a listing of integers to be build representing our
vertex coordinates in 2D in list fashion.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="OffScreenGeometry2D" type="OffScreenGeometry2DType">
  <xsd:annotation>
    <xsd:documentation>Consists of a declaration for number of vertices and a corresponding listing of tuple
values in order to build basic off-screen representations for geography.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="StringsURL" type="StringsUrl">
  <xsd:annotation>
    <xsd:documentation>From the x3d schema, one method of representing a string based url in a list of
strings.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!--Entity View type declarations go here...not essential but nice refs for anyone looking to use that doesn't

```

```

know Java.-->
  <xsd:complexType name="Primary3DViewType">
    <xsd:annotation>
      <xsd:documentation>Type representing primary 3d view url or file reference.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
      <xsd:element ref="StringsURL"/>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="Secondary3DViewType">
    <xsd:annotation>
      <xsd:documentation>Secondary 3D View Type represents the secondary 3D View for an
entity.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
      <xsd:element ref="StringsURL"/>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="Miscellaneous3DViewType">
    <xsd:annotation>
      <xsd:documentation>Miscellaneous 3DView Type represents a miscellaneous 3D View for an entity tertiary in
priority and greater.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
      <xsd:element ref="StringsURL"/>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="Primary2DViewType">
    <xsd:annotation>
      <xsd:documentation>Primary 2D ViewType represents the primary 2D View type for an
entity.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
      <xsd:element ref="StringsURL"/>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="Secondary2DViewType">
    <xsd:annotation>
      <xsd:documentation>Secondary2DViewType represents the secondary 2D View type for an
entity.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>

```

```

        <xsd:element ref="StringsURL"/>
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="Miscellaneous2DViewType">
    <xsd:annotation>
        <xsd:documentation>Miscellaneous 2DViewType is a general purpose storage for tertiary and greater 2D view
references.</xsd:documentation>
    </xsd:annotation>
    <xsd:all>
        <xsd:element ref="StringsURL"/>
    </xsd:all>
</xsd:complexType>
<xsd:element name="Primary3DView" type="Primary3DViewType">
    <xsd:annotation>
        <xsd:documentation>Element representing primary 3d view url or file reference.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Secondary3DView" type="Secondary3DViewType">
    <xsd:annotation>
        <xsd:documentation>Secondary 3D View element can store the secondary 3D view for an
entity.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Miscellaneous3DView" type="Miscellaneous3DViewType">
    <xsd:annotation>
        <xsd:documentation>Miscellaneous3DView element can store alternate file or url references for renderable
geometry if there are more than two in use.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Primary2DView" type="Primary2DViewType">
    <xsd:annotation>
        <xsd:documentation>Primary 2D View element contains the string based url or file location for this 2D
view.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Secondary2DView" type="Secondary2DViewType">
    <xsd:annotation>
        <xsd:documentation>Secondary 2D View element contains the string based url or file location for this 2D
view.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="Miscellaneous2DView" type="Miscellaneous2DViewType">

```

```

        <xsd:annotation>
            <xsd:documentation>Misscellaneous 2D Viewelement contains the string based url or file location for this 2D
view.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="ViewPropertyType">
        <xsd:annotation>
            <xsd:documentation>EntityViewType is a storage bin for various entity views (ie 3D, 2D jpeg, 2D svg,
etc).</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="Primary3DView"/>
            <xsd:element ref="Secondary3DView" minOccurs="0"/>
            <xsd:element ref="Miscellaneous3DView" minOccurs="0"/>
            <xsd:element ref="Primary2DView" minOccurs="0"/>
            <xsd:element ref="Secondary2DView" minOccurs="0"/>
            <xsd:element ref="Miscellaneous2DView" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="ViewProperties" type="ViewPropertyType">
        <xsd:annotation>
            <xsd:documentation>Represents files or url's for view references for an entity.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <!--DIS Specific Information Here. Network Properties setup so at least the Network controls are agnostic to
protocol format.-->
    <xsd:complexType name="DISInformationType">
        <xsd:annotation>
            <xsd:documentation>DISInformation Type is the DIS specific information for this entity.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="EntityIdentification"/>
            <xsd:element ref="SiteIdentification"/>
            <xsd:element ref="ApplicationIdentification"/>
            <xsd:element ref="EntityKind"/>
            <xsd:element ref="EntityCountry"/>
            <xsd:element ref="EntityDomain"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="DISInformation" type="DISInformationType">
        <xsd:annotation>
            <xsd:documentation>DISInformation Element is the DIS specific information element for this

```



```

entity.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="XYZPositionType">
  <xsd:annotation>
    <xsd:documentation>Represents an entities position in Graphics Space</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="GISStringAnchorLatitudeType">
  <xsd:annotation>
    <xsd:documentation>Represents Latittud anchor point that is equivalent to 0,0,0 in the 3D
view.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="GISStringAnchorLongitudeType">
  <xsd:annotation>
    <xsd:documentation>Represents Longitude anchor point that is equivalent to 0,0,0 in the 3D
view.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="XYZPosition" type="XYZPositionType">
  <xsd:annotation>
    <xsd:documentation>XYZPosition represents the triplet location value for use in 3D
space.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="GISStringAnchorLatitude" type="GISStringAnchorLatitudeType">
  <xsd:annotation>
    <xsd:documentation>GISStringAnchorLatitude element represents a latitude point to anchor a 3D coordinate
system on typically representing the position 0 0 0.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="GISStringAnchorLongitude" type="GISStringAnchorLongitudeType">
  <xsd:annotation>
    <xsd:documentation>GISStringAnchorLongitude element represents a longitude point to anchor a 3D coordinate
system on with it representing the 0 0 0 point in 3D generally.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="GISStringLatitudePositionType">

```

```

        <xsd:annotation>
            <xsd:documentation>GISStringLatitudePosition Type represents a basic string type for representing latitude
positions.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:simpleType name="GISStringLongitudePositionType">
        <xsd:annotation>
            <xsd:documentation>GISStrinLongitudePositionType represents a basic string type for representing longitude
positions.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:string"/>
    </xsd:simpleType>
    <xsd:element name="GISStringLatitudePosition" type="GISStringLatitudePositionType">
        <xsd:annotation>
            <xsd:documentation>GISStringLatittudePosition element represents a position in GIS latitude
coordinates.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="GISStringLongitudePosition" type="GISStringLongitudePositionType">
        <xsd:annotation>
            <xsd:documentation>GISStringLongitudePosition element represents a position in GIS longitude coordinates as
a string type.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="EntityPositionType">
        <xsd:annotation>
            <xsd:documentation>Entity Position type contains all location information for an entity whether GIS, 3D
Space or both.</xsd:documentation>
        </xsd:annotation>
        <xsd:choice>
            <xsd:sequence>
                <xsd:element ref="XYZPosition"/>
                <xsd:element ref="GISStringAnchorLatitude"/>
                <xsd:element ref="GISStringAnchorLongitude"/>
                <xsd:element ref="XYPosition" minOccurs="0"/>
            </xsd:sequence>
            <xsd:sequence>
                <xsd:element ref="GISStringLatitudePosition" minOccurs="0"/>
                <xsd:element ref="GISStringLongitudePosition" minOccurs="0"/>
            </xsd:sequence>
        </xsd:choice>
    </xsd:complexType>

```

```

</xsd:complexType>
<xsd:element name="EntityPosition" type="EntityPositionType">
  <xsd:annotation>
    <xsd:documentation>Entity Position element contains all location information for an
entity.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="LocationPosition" type="EntityPositionType">
  <xsd:annotation>
    <xsd:documentation>Defines any location offsets needed to render in the applicable location for the defined
geography.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="EntityRateOfTurnType">
  <xsd:annotation>
    <xsd:documentation>Represents the rate of turn about the Y or up axis in radians.</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="EntityRateOfTurn" type="EntityRateOfTurnType">
  <xsd:annotation>
    <xsd:documentation>Element that represents the rate of turn for an entity about the Y
axis.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="EntityHeadingType">
  <xsd:annotation>
    <xsd:documentation>Entity heading represented by the tuple x rot, y rot, z rot, amount of turn in
radians</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="EntityHeading" type="EntityHeadingType">
  <xsd:annotation>
    <xsd:documentation>Element representing an entities heading as a string with the format x rot, y rot, z
rot, degrees of rotation in radians.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="IdentificationRangeType">
  <xsd:annotation>
    <xsd:documentation>IdentificationRangeType represents a integer based range for which to identify targets
by.</xsd:documentation>

```

```

        </xsd:annotation>
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="InterceptRangeType">
        <xsd:annotation>
            <xsd:documentation>InterceptRangeType represents an intercept based range for which to intercept targets
by.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name="EngagementRangeType">
        <xsd:annotation>
            <xsd:documentation>Engagement Range TYPe represents an engagement range type to engage targets
by.</xsd:documentation>
        </xsd:annotation>
        <xsd:restriction base="xsd:integer"/>
    </xsd:simpleType>
    <xsd:element name="IdentificationRange" type="IdentificationRangeType">
        <xsd:annotation>
            <xsd:documentation>Identification Range is an element that represents when an entity wants to id someone
within this threshold.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="InterceptRange" type="InterceptRangeType">
        <xsd:annotation>
            <xsd:documentation>Intercept Range is an element that represents when an entity wants to intercept
something by.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="EngagementRange" type="EngagementRangeType">
        <xsd:annotation>
            <xsd:documentation>Engagement Range is an element that represents when we want to engage
something.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="DefenseParameter" type="DefenseParameterType">
        <xsd:annotation>
            <xsd:documentation>Contains a 3 tuple for Identification Range, Intercept Range, and Engagement ranges for
an entity that needs these.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:simpleType name="ValueType">

```

```

    <xsd:annotation>
      <xsd:documentation>Value Type represents a generalized value element that can store any simple
type.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:anySimpleType"/>
  </xsd:simpleType>
  <xsd:element name="Value" type="ValueType">
    <xsd:annotation>
      <xsd:documentation>Value element represents a generalized value that can store any simple
element.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:simpleType name="XYPositionType">
    <xsd:annotation>
      <xsd:documentation>XYPosition type represents an entities position in 2DCoordinates.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
  <xsd:element name="XYPosition" type="XYPositionType"/>
</xsd:schema>

```

APPENDIX B. XSLT STYLESHEET FOR ENTITY PROPERTIES AND TACTICAL DATA

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by Khaled Mnif(US Naval Postgraduate School) -->
<!--W3C Schema generated by XML Spy v4.3 (http://www.xmlspy.com) -->
<!--Authors: James Harney, Khaled Mnif-->
<!--Created: 5 November 2002-->
<!--Revised: 20 December 2002-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head/>
      <body>
        <xsl:for-each select="ATFPScenarioType">
          <xsl:for-each select="Head">
            <xsl:for-each select="TacticalData">
              <br/>
              <table align="center" bgcolor="#0000A0" border="3" frame="box" width="100%">
                <xsl:if test="position()=1">
                  <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
                </xsl:if>
                <tr>
                  <td align="center" height="51">
                    <span style="color:#FFFFFF; font-size:large; font-style:italic; font-weight:bold">ATFP
Scenario Entities:</span>
                  </td>
                </tr>
                <xsl:if test="position()=last()">
                  <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
                </xsl:if>
              </table>
              <br/>
              <xsl:for-each select="Entity">
                <table bgcolor="#7171FF" border="0" width="100%">
                  <xsl:if test="position()=1">
                    <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
                  </xsl:if>
                  <tr align="left">
                    <td height="36">
```

```

weight:bold">Entity</span>
weight:bold">:..... </span>
weight:bold">
    <xsl:for-each select="EntityProperties">
        <xsl:for-each select="Name">
            <span style="color:#B30059; font-size:medium; font-style:normal; font-
                <xsl:apply-templates/>
            </span>
        </xsl:for-each>
    </xsl:for-each>
</td>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
<br/>
<xsl:for-each select="EntityProperties">
    <span style="font-weight:bold; text-decoration:underline">Properties:</span>
    <br/>
    <xsl:if test="position()=1">
        <xsl:text disable-output-escaping="yes">&lt;table border="0"&gt;</xsl:text>
    </xsl:if>
    <xsl:if test="position()=1">
        <thead>
            <tr bgcolor="#FFFF80">
                <td align="center" height="34">Identification</td>
                <td align="center" height="34">Length</td>
                <td align="center" height="34">Width</td>
                <td align="center" height="34">Height</td>
                <td align="center" height="34">Position</td>
                <td align="center" height="34">Rate Of Turn</td>
                <td align="center" height="34">Heading</td>
            </tr>
        </thead>
    </xsl:if>
    <xsl:if test="position()=1">
        <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
    </xsl:if>
    <tr align="center" bgcolor="#FFBB77">

```

```

<td align="center" height="39">
  <xsl:for-each select="Identification">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="Length">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="Width">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="Height">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="EntityPosition">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="EntityRateOfTurn">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
<td height="39">
  <xsl:for-each select="EntityHeading">
    <xsl:apply-templates/>
  </xsl:for-each>
</td>
</tr>
<xsl:if test="position()=last()">
  <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
<xsl:if test="position()=last()">
  <xsl:text disable-output-escaping="yes">&lt;/table&gt;</xsl:text>
</xsl:if>

```



```

</xsl:for-each>
<br/>
<span style="font-weight:bold; text-decoration:underline">Tactical Data:</span>
<br/>
<xsl:for-each select="TacticalProperties">
  <xsl:if test="position()=1">
    <xsl:text disable-output-escaping="yes">&lt;table border="0"&gt;</xsl:text>
  </xsl:if>
  <xsl:if test="position()=1">
    <thead>
      <tr align="center" bgcolor="#88FF88">
        <td height="46">Weapon Properties</td>
        <td height="46">Line Of Site Distance</td>
        <td height="46">Line Of Site Probability Of Siting</td>
        <td height="46">General Property</td>
        <td height="46">Defense Parameter</td>
      </tr>
    </thead>
  </xsl:if>
  <xsl:if test="position()=1">
    <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
  </xsl:if>
  <tr align="center" bgcolor="#00DB37">
    <td>
      <xsl:for-each select="WeaponProperties">
        <xsl:if test="position()=1">
          <table border="1">
            <xsl:if test="position()=1">
              <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
            </xsl:if>
            <tr>
              <td align="justify" bgcolor="#80FF80" width="127">Probability Success</td>
              <xsl:for-each select="../WeaponProperties">
                <td width="175">
                  <xsl:for-each select="ProbabilityOfWeaponSuccess">
                    <span style="color:#000000; font-weight:normal">
                      <xsl:apply-templates/>
                    </span>
                  </xsl:for-each>
                </td>
              </xsl:for-each>
            </tr>
          </table>
        </xsl:if>
      </xsl:for-each>
    </td>
  </tr>

```

```

<tr>
  <td align="justify" bgcolor="#80FF80" width="127">Name</td>
  <xsl:for-each select=" ../WeaponProperties">
    <td width="175">
      <xsl:for-each select="WeaponName">
        <span style="color:#000000; font-weight:normal">
          <xsl:apply-templates/>
        </span>
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td align="justify" bgcolor="#80FF80" width="127">Model</td>
  <xsl:for-each select=" ../WeaponProperties">
    <td width="175">
      <xsl:for-each select="WeaponModel">
        <span style="color:black; font-weight:normal">
          <xsl:apply-templates/>
        </span>
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td align="justify" bgcolor="#80FF80" width="127">Amount</td>
  <xsl:for-each select=" ../WeaponProperties">
    <td width="175">
      <xsl:for-each select="WeaponAmount">
        <span style="color:black; font-weight:normal">
          <xsl:apply-templates/>
        </span>
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td align="justify" bgcolor="#80FF80" width="127">Lethality Range</td>
  <xsl:for-each select=" ../WeaponProperties">
    <td width="175">
      <xsl:for-each select="WeaponLethalityRange">
        <span style="color:black; font-weight:normal">

```

```

                <xsl:apply-templates/>
            </span>
        </xsl:for-each>
    </td>
</xsl:for-each>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
</xsl:if>
</xsl:for-each>
</td>
<td>
    <xsl:for-each select="LineOfSiteDistance">
        <span style="color:black; font-weight:normal">
            <xsl:apply-templates/>
        </span>
    </xsl:for-each>
</td>
<td>
    <xsl:for-each select="LineOfSiteProbabilityOfSiting">
        <span style="color:black; font-weight:normal">
            <xsl:apply-templates/>
        </span>
    </xsl:for-each>
</td>
<td>
    <xsl:for-each select="GeneralProperty">
        <span style="color:black; font-weight:normal">
            <xsl:apply-templates/>
        </span>
    </xsl:for-each>
</td>
<td>
    <xsl:for-each select="DefenseParameter">
        <xsl:if test="position()=1">
            <table border="1" frame="border">
                <xsl:if test="position()=1">
                    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
                </xsl:if>
            <tr>

```

```

<td bgcolor="#80FF80" height="44">Identification Range</td>
<xsl:for-each select=" ../DefenseParameter">
  <td height="44">
    <xsl:for-each select="IdentificationRange">
      <span style="color:black; font-weight:normal">
        <xsl:apply-templates/>
      </span>
    </xsl:for-each>
  </td>
</xsl:for-each>
</tr>
<tr>
  <td bgcolor="#80FF80" height="38">Intercept Range</td>
  <xsl:for-each select=" ../DefenseParameter">
    <td height="38">
      <xsl:for-each select="InterceptRange">
        <span style="color:black; font-weight:normal">
          <xsl:apply-templates/>
        </span>
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td bgcolor="#80FF80" height="39">Engagement Range</td>
  <xsl:for-each select=" ../DefenseParameter">
    <td height="39">
      <xsl:for-each select="EngagementRange">
        <span style="color:black; font-weight:normal">
          <xsl:apply-templates/>
        </span>
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<xsl:if test="position()=last()">
  <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
</xsl:if>
</xsl:for-each>
</td>

```

```

        </tr>
        <xsl:if test="position()=last()">
            <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
        </xsl:if>
        <xsl:if test="position()=last()">
            <xsl:text disable-output-escaping="yes">&lt;/table&gt;</xsl:text>
        </xsl:if>
    </xsl:for-each>
    <br/>
</xsl:for-each>
<br/>
<table bgcolor="#0000A0" border="3" frame="box" width="100%">
    <xsl:if test="position()=1">
        <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
    </xsl:if>
    <tr align="center" bgcolor="#000080">
        <td height="49">
            <span style="color:#FFFFFF; font-size:large; font-style:italic; font-weight:bold">AFTP
Scenario Location:</span>
        </td>
    </tr>
    <xsl:if test="position()=last()">
        <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
    </xsl:if>
</table>
<br/>
<span style="font-weight:bold; text-decoration:underline">Description:</span>
<br/>
<xsl:for-each select="Location">
    <xsl:if test="position()=1">
        <xsl:text disable-output-escaping="yes">&lt;table border="0"&gt;</xsl:text>
    </xsl:if>
    <xsl:if test="position()=1">
        <thead>
            <tr align="center" bgcolor="#BE7DFF">
                <td height="44">Name</td>
                <td height="44">Upper Left Geo. Position</td>
                <td height="44">Lower Right Geo. Position</td>
                <td height="44">Position</td>
            </tr>
        </thead>
    </xsl:if>

```

```

<xsl:if test="position()=1">
  <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
</xsl:if>
<tr bgcolor="#8F20FF">
  <td height="40">
    <xsl:for-each select="LocationName">
      <a href="C:\Documents and
Settings\khaled\Desktop\Thesis\MyWork\SenariosObjects\PortHuenemePiersNPSNET.wrl"/>
      <xsl:apply-templates/>
    </xsl:for-each>
  </td>
  <td height="40">
    <xsl:for-each select="LocationUpperLeftGeoPosition">
      <xsl:apply-templates/>
    </xsl:for-each>
  </td>
  <td height="40">
    <xsl:for-each select="LocationLowerRightGeoPosition">
      <xsl:apply-templates/>
    </xsl:for-each>
  </td>
  <td height="40">
    <xsl:for-each select="LocationPosition">
      <xsl:apply-templates/>
    </xsl:for-each>
  </td>
</tr>
<xsl:if test="position()=last()">
  <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
<xsl:if test="position()=last()">
  <xsl:text disable-output-escaping="yes">&lt;/table&gt;</xsl:text>
</xsl:if>
</xsl:for-each>
<br/>
<br/>
<br/>
<br/>
<br/>
</xsl:for-each>
</xsl:for-each>
</xsl:for-each>

```

```

    </body>
</html>
</xsl:template>
<xsl:template match="Entity">
  <xsl:for-each select="Entity">
    <xsl:if test="position()=1">
      <table border="1">
        <xsl:if test="position()=1">
          <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
        </xsl:if>
        <tr>
          <td>EntityProperties</td>
          <xsl:for-each select="../Entity">
            <td>
              <xsl:for-each select="EntityProperties">
                <xsl:for-each select="EntityProperties">
                  <xsl:if test="position()=1">
                    <table border="1">
                      <xsl:if test="position()=1">
                        <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
                      </xsl:if>
                      <tr>
                        <td>Name</td>
                        <xsl:for-each select="../EntityProperties">
                          <td>
                            <xsl:for-each select="Name">
                              <xsl:apply-templates/>
                            </xsl:for-each>
                          </td>
                        </xsl:for-each>
                      </tr>
                      <tr>
                        <td>Identification</td>
                        <xsl:for-each select="../EntityProperties">
                          <td>
                            <xsl:for-each select="Identification">
                              <xsl:apply-templates/>
                            </xsl:for-each>
                          </td>
                        </xsl:for-each>
                      </tr>
                      <tr>

```

```

        <td>Length</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="Length">
                    <xsl:apply-templates/>
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>Width</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="Width">
                    <xsl:apply-templates/>
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>Height</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="Height">
                    <xsl:apply-templates/>
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>MinimumSpeed</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="MinimumSpeed">
                    <xsl:apply-templates/>
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>MaximumSpeed</td>
        <xsl:for-each select="../EntityProperties">

```



```

        <td>
            <xsl:for-each select="MaximumSpeed">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>GeneralProperty</td>
    <xsl:for-each select="../EntityProperties">
        <td>
            <xsl:for-each select="GeneralProperty">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>EntityPosition</td>
    <xsl:for-each select="../EntityProperties">
        <td>
            <xsl:for-each select="EntityPosition">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>EntityRateOfTurn</td>
    <xsl:for-each select="../EntityProperties">
        <td>
            <xsl:for-each select="EntityRateOfTurn"><xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>EntityHeading</td>
    <xsl:for-each select="../EntityProperties">
        <td>
            <xsl:for-each select="EntityHeading">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>

```

```

                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <xsl:if test="position()=last()">
        <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
    </xsl:if>
</table>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
</td>
</xsl:for-each>
</tr>
<tr>
    <td>TacticalProperties</td>
    <xsl:for-each select="../Entity">
        <td>
            <xsl:for-each select="TacticalProperties"><xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>ViewProperties</td>
    <xsl:for-each select="../Entity">
        <td>
            <xsl:for-each select="ViewProperties">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
</xsl:if>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. XSLT STYLESHEET FOR MESURE OF EFFECTIVENESS

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY Designer (http://www.xmlspy.com) by Khaled Mnif (US Naval Postgraduate School) -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head />
      <body>
        <xsl:for-each select="ATFPScenarioType">
          <xsl:for-each select="Head">
            <br />
            <br />
            <br />
            <table bgcolor="#800000" border="0" width="100%">
              <xsl:if test="position()=1">
                <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
              </xsl:if>
              <tr align="center">
                <td height="60">
                  <span style="color:#FFFF00; font-size:large; font-weight:bold">ATFP Scenario
Statistic Output</span>
                </td>
              </tr>
              <xsl:if test="position()=last()">
                <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
              </xsl:if>
            </table>
            <br />
            <xsl:for-each select="StatisticOutput">
              <br />
              <table bgcolor="#000080" border="0" frame="box" width="100%">
                <xsl:if test="position()=1">
                  <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
                </xsl:if>
                <tr>
                  <td align="center" height="42">
                    <span style="color:yellow; font-size:large; font-style:italic; font-
weight:bold"> </span>
```

```

        <span style="color:yellow; font-size:large; font-style:italic; font-
weight:bold">The Overall Measure Of Effectiveness</span>
    </td>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
<br />
<xsl:for-each select="MOE">
    <xsl:for-each select="OverAllScenarios">

        <xsl:if test="position()=1">
            <xsl:text disable-output-escaping="yes">&lt;table align="center"
border="0"&gt;</xsl:text>
        </xsl:if>
        <xsl:if test="position()=1">
            <thead>
                <tr align="center" bgcolor="#FFFF91">
                    <td>Total Number Runs</td>
                    <td>Random Number Seed</td>
                    <td>Defense Percent Effective</td>
                    <td>Mean Percent Effective</td>
                    <td>Standard Deviation</td>
                    <td>Variance</td>
                </tr>
            </thead>
        </xsl:if>
        <xsl:if test="position()=1">
            <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
        </xsl:if>
        <tr align="center" bgcolor="#B7B7FF">
            <td height="47">
                <xsl:for-each select="@TotalNumberRuns">
                    <xsl:value-of select="." />
                </xsl:for-each>
            </td>
            <td height="47">
                <xsl:for-each select="@RandomNumberSeed">
                    <xsl:value-of select="." />
                </xsl:for-each>
            </td>

```

```

        <td height="47">
            <xsl:for-each select="@DefensePercentEffective">
                <xsl:value-of select="." />
            </xsl:for-each>
        </td>
        <td height="47">
            <xsl:for-each select="@MeanPercentEffective">
                <xsl:value-of select="." />
            </xsl:for-each>
        </td>
        <td height="47">
            <xsl:for-each select="@StandardDeviation">
                <xsl:value-of select="." />
            </xsl:for-each>
        </td>
        <td height="47">
            <xsl:for-each select="@Variance">
                <xsl:value-of select="." />
            </xsl:for-each>
        </td>
    </tr>
    <xsl:if test="position()=last()">
        <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
    </xsl:if>
    <xsl:if test="position()=last()">
        <xsl:text disable-output-escaping="yes">&lt;/table&gt;</xsl:text>
    </xsl:if>
    <br />
</xsl:for-each>
</xsl:for-each>
<br />
<table border="0" width="100%">
    <xsl:if test="position()=1">
        <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
    </xsl:if>
    <tr align="center" bgcolor="#000080">
        <td height="47">
            <span style="color:#FFFF00; font-size:large; font-style:italic; font-
weight:bold">Measure of Effectiveness by Scenario Run</span>
        </td>
    </tr>
    <xsl:if test="position()=last()">

```

```

        <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
    </xsl:if>
</table>
<br />
<br />
<xsl:for-each select="MOE">
    <xsl:for-each select="ScenarioRun">
        <xsl:if test="position()=1">
            <xsl:text disable-output-escaping="yes">&lt;table align="center"
border="0"&gt;</xsl:text>

            </xsl:if>
            <xsl:if test="position()=1">
                <thead>
                    <tr align="center" bgcolor="#808080">
                        <td height="42">Scenario Run No</td>
                        <td height="42">Defense Percent Effective</td>
                        <td height="42">Mean Percent Effective</td>
                        <td height="42">Standard Deviation</td>
                        <td height="42">Variance</td>
                    </tr>
                </thead>
            </xsl:if>
            <xsl:if test="position()=1">
                <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
            </xsl:if>
            <tr align="center" bgcolor="#C0C0C0">
                <td height="33">
                    <xsl:for-each select="@value">
                        <xsl:value-of select="." />
                    </xsl:for-each>
                </td>
                <td height="33">
                    <xsl:for-each select="DefensePercentEffective">
                        <xsl:for-each select="@value">
                            <xsl:value-of select="." />
                        </xsl:for-each>
                    </xsl:for-each>
                </td>
                <td height="33">
                    <xsl:for-each select="MeanPercentEffective">
                        <xsl:for-each select="@value">
                            <xsl:value-of select="." />

```

```

        </xsl:for-each>
        </xsl:for-each>
    </td>
    <td height="33">
        <xsl:for-each select="StandardDeviation">
            <xsl:for-each select="@value">
                <xsl:value-of select="." />
            </xsl:for-each>
        </xsl:for-each>
    </td>
    <td height="33">
        <xsl:for-each select="Variance">
            <xsl:for-each select="@value">
                <xsl:value-of select="." />
            </xsl:for-each>
        </xsl:for-each>
    </td>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/table&gt;</xsl:text>
</xsl:if>
</xsl:for-each>
</xsl:for-each>
<br />
<br />
<br />
<br />
</xsl:for-each>
</xsl:for-each>
</xsl:for-each>
</body>
</html>
</xsl:template>
<xsl:template match="Entity">
    <xsl:for-each select="Entity">
        <xsl:if test="position()=1">
            <table border="1">
                <xsl:if test="position()=1">
                    <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>

```



```

</xsl:if>
<tr>
  <td>EntityProperties</td>
  <xsl:for-each select="../Entity">
    <td>
      <xsl:for-each select="EntityProperties">
        <xsl:for-each select="EntityProperties">
          <xsl:if test="position()=1">
            <table border="1">
              <xsl:if test="position()=1">
                <xsl:text disable-output-escaping="yes">&lt;tbody&gt;</xsl:text>
              </xsl:if>
              <tr>
                <td>Name</td>
                <xsl:for-each select="../EntityProperties">
                  <td>
                    <xsl:for-each select="Name">
                      <xsl:apply-templates />
                    </xsl:for-each>
                  </td>
                </xsl:for-each>
              </tr>
            </table>
          </xsl:if>
          <tr>
            <td>Identification</td>
            <xsl:for-each select="../EntityProperties">
              <td>
                <xsl:for-each select="Identification">
                  <xsl:apply-templates />
                </xsl:for-each>
              </td>
            </xsl:for-each>
          </tr>
          <tr>
            <td>Length</td>
            <xsl:for-each select="../EntityProperties">
              <td>
                <xsl:for-each select="Length">
                  <xsl:apply-templates />
                </xsl:for-each>
              </td>
            </xsl:for-each>
          </tr>
        </xsl:for-each>
      </td>
    </xsl:for-each>
  </tr>

```

```

<tr>
  <td>Width</td>
  <xsl:for-each select="../EntityProperties">
    <td>
      <xsl:for-each select="Width">
        <xsl:apply-templates />
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td>Height</td>
  <xsl:for-each select="../EntityProperties">
    <td>
      <xsl:for-each select="Height">
        <xsl:apply-templates />
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td>MinimumSpeed</td>
  <xsl:for-each select="../EntityProperties">
    <td>
      <xsl:for-each select="MinimumSpeed">
        <xsl:apply-templates />
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td>MaximumSpeed</td>
  <xsl:for-each select="../EntityProperties">
    <td>
      <xsl:for-each select="MaximumSpeed">
        <xsl:apply-templates />
      </xsl:for-each>
    </td>
  </xsl:for-each>
</tr>
<tr>
  <td>GeneralProperty</td>

```

```

        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="GeneralProperty">
                    <xsl:apply-templates />
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>EntityPosition</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="EntityPosition">
                    <xsl:apply-templates />
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>EntityRateOfTurn</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="EntityRateOfTurn">
                    <xsl:apply-templates />
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <tr>
        <td>EntityHeading</td>
        <xsl:for-each select="../EntityProperties">
            <td>
                <xsl:for-each select="EntityHeading">
                    <xsl:apply-templates />
                </xsl:for-each>
            </td>
        </xsl:for-each>
    </tr>
    <xsl:if test="position()=last()">
        <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
    </xsl:if>
</table>

```

```

        </xsl:if>
        </xsl:for-each>
    </xsl:for-each>
</td>
</xsl:for-each>
</tr>
<tr>
    <td>TacticalProperties</td>
    <xsl:for-each select="../Entity">
        <td>
            <xsl:for-each select="TacticalProperties">
                <xsl:apply-templates />
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td>ViewProperties</td>
    <xsl:for-each select="../Entity">
        <td>
            <xsl:for-each select="ViewProperties">
                <xsl:apply-templates />
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<xsl:if test="position()=last()">
    <xsl:text disable-output-escaping="yes">&lt;/tbody&gt;</xsl:text>
</xsl:if>
</table>
</xsl:if>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. AT/FP SCENARIO INSTANCE

```
<?xml version="1.0" encoding="UTF-8"?>
<ATFPScenarioType>
  <meta content="filename" name="ATFPNewResult.xml"/>
  <Head>
    <TacticalData>
      <Location>
        <LocationName>PortHueneme</LocationName>
        <LocationUpperLeftGeoPosition>34 15.00N 119 00W</LocationUpperLeftGeoPosition>
        <LocationLowerRightGeoPosition>34 00.00N 119 15.00W</LocationLowerRightGeoPosition>
        <OffScreenGeometry2D>
          <VerticeNumber>17</VerticeNumber>
          <IntegerList>5551,5811,5811,5701,5741,5811,5841,5841,6011,6011,6211,6211,6571,6571,6491,5901,5701,6602,
            6262,5812,5722,5702,5752,5752,5732,5732,5902,5902,6242,6242,6302,6382,6382,6622
          </IntegerList>
        </OffScreenGeometry2D>
        <ViewProperties>
          <Primary3DView>
            <StringsURL>C:\atfp/resources/org/npsnet/v/views/terrain/PortHuenemePiers.wrl</StringsURL>
          </Primary3DView>
          <Secondary3DView>
            <StringsURL>http://web.nps.navy.mil/~brutzman/Savage/Locations/PortHuenemeCalifornia/PortHuenemePiers.wrl
          </StringsURL>
          </Secondary3DView>
          <Primary2DView>
            <StringsURL>C:\atfp/images/PortHuenemeChart.gif</StringsURL>
          </Primary2DView>
          <Secondary2DView>
            <StringsURL>jar:file:/C:\atfp/images/images.jar!/PortHuenemeChart.gif</StringsURL>
          </Secondary2DView>
        </ViewProperties>
        <LocationPosition>
          <XYZPosition>5551 0 6602</XYZPosition>
          <GISStringAnchorLatitude>34 00.0N</GISStringAnchorLatitude>
          <GISStringAnchorLongitude>119 00W</GISStringAnchorLongitude>
          <XYPosition>0 0</XYPosition>
        </LocationPosition>
      </Location>
    </TacticalData>
  </Head>
</ATFPScenarioType>
```

```

<Entity>
  <EntityProperties>
    <Name>CG-47</Name>
    <Identification>Ticonderoga Class Cruiser</Identification>
    <Length>162.0</Length>
    <Width>13.5</Width>
    <Height>20.0</Height>
    <MinimumSpeedElement>-5.0</MinimumSpeedElement>
    <MaximumSpeedElement>30.0</MaximumSpeedElement>
    <EntityPosition>
      <XYZPosition>6186.0 0 6152.2</XYZPosition>
      <GISStringAnchorLatitude>34 00.0N</GISStringAnchorLatitude>
      <GISStringAnchorLongitude>119 00W</GISStringAnchorLongitude>
      <XYPosition>641.0 223.0</XYPosition>
    </EntityPosition>
    <EntityRateOfTurn>0.1058</EntityRateOfTurn>
    <EntityHeading>0.0</EntityHeading>
  </EntityProperties>
  <ControlModeProperties>Agent Controlled</ControlModeProperties>
  <NetworkProperties>
    <NetworkProtocolPrimary>DIS</NetworkProtocolPrimary>
    <IPAddress>127.0.0.1</IPAddress>
    <DISInformation>
      <EntityIdentification>67</EntityIdentification>
      <SiteIdentification>0</SiteIdentification>
      <ApplicationIdentification>1</ApplicationIdentification>
      <EntityKind>1</EntityKind>
      <EntityCategory>3</EntityCategory>
      <EntityDomain>3</EntityDomain>
    </DISInformation>
  </NetworkProperties>
  <TacticalProperties>
    <WeaponProperties>
      <ProbabilityOfWeaponSuccess>80.0</ProbabilityOfWeaponSuccess>
      <WeaponName>Small Arms</WeaponName>
      <WeaponModel>Various</WeaponModel>
      <WeaponAmount>0.0</WeaponAmount>
      <WeaponLethalityRange>100.0</WeaponLethalityRange>
    </WeaponProperties>
    <LineOfSiteDistance>6000.0</LineOfSiteDistance>
    <LineOfSiteProbabilityOfSiting>1.0</LineOfSiteProbabilityOfSiting>
    <DefenseParameter>

```

```

        <IdentificationRange>650</IdentificationRange>
        <InterceptRange>500</InterceptRange>
        <EngagementRange>250</EngagementRange>
    </DefenseParameter>
</TacticalProperties>
<ViewProperties>
    <Primary3DView>

    <StringsURL>C:\atfp/resources/org/npsnet/v/views/platforms/surface/CruiserTiconderoga.wrl</StringsURL>
    </Primary3DView>
    <Secondary3DView>
        <StringsURL>http://web.nps.navy.mil/~brutzman/Savage/Ships/Cruiser-
UnitedStates/Cruiser.wrl</StringsURL>
    </Secondary3DView>
    <Primary2DView>
        <StringsURL>C:\atfp/images/ddgIcon.gif</StringsURL>
    </Primary2DView>
    <Secondary2DView>
        <StringsURL>jar:file:/C:\atfp/images/images.jar!/ddgIcon.gif</StringsURL>
    </Secondary2DView>
</ViewProperties>
</Entity>
<Entity>
    <EntityProperties>
        <Name>rhjb</Name>
        <Identification>Defensive Picket Boat</Identification>
        <Length>20.0</Length>
        <Width>3.0</Width>
        <Height>1.0</Height>
        <MinimumSpeedElement>-5.0</MinimumSpeedElement>
        <MaximumSpeedElement>30.0</MaximumSpeedElement>
        <EntityPosition>
            <XYZPosition>5973.8 0 6149.6</XYZPosition>
            <GISStringAnchorLatitude>34 00.0N</GISStringAnchorLatitude>
            <GISStringAnchorLongitude>119 00W</GISStringAnchorLongitude>
            <XYPosition>586.0 276.0</XYPosition>
        </EntityPosition>
        <EntityRateOfTurn>0.1058</EntityRateOfTurn>
        <EntityHeading>3.3161255787892263</EntityHeading>
    </EntityProperties>
    <ControlModeProperties>Agent Controlled</ControlModeProperties>
    <NetworkProperties>

```



```

    <NetworkProtocolPrimary>DIS</NetworkProtocolPrimary>
    <IPAddress>127.0.0.1</IPAddress>
    <DISInformation>
        <EntityIdentification>66</EntityIdentification>
        <SiteIdentification>0</SiteIdentification>
        <ApplicationIdentification>1</ApplicationIdentification>
        <EntityKind>1</EntityKind>
        <EntityCategory>0</EntityCategory>
        <EntityDomain>0</EntityDomain>
    </DISInformation>
</NetworkProperties>
<TacticalProperties>
    <WeaponProperties>
        <ProbabilityOfWeaponSuccess>70.0</ProbabilityOfWeaponSuccess>
        <WeaponName>Intercept</WeaponName>
        <WeaponModel>Target Intercept</WeaponModel>
        <WeaponAmount>0.0</WeaponAmount>
        <WeaponLethalityRange>15.0</WeaponLethalityRange>
    </WeaponProperties>
    <LineOfSiteDistance>5000.0</LineOfSiteDistance>
    <LineOfSiteProbabilityOfSiting>1.0</LineOfSiteProbabilityOfSiting>
</TacticalProperties>
<ViewProperties>
    <Primary3DView>

<StringsURL>C:\atfp/resources/org/npsnet/v/views/platforms/surface/RhibPrototypeNPSNET.wrl</StringsURL>
    </Primary3DView>
    <Secondary3DView>
        <StringsURL>http://web.nps.navy.mil/~brutzman/Savage/Scenarios/RhibPrototype.wrl</StringsURL>
    </Secondary3DView>
    <Primary2DView>
        <StringsURL>C:\atfp/images/RHIBIcon.gif</StringsURL>
    </Primary2DView>
    <Secondary2DView>
        <StringsURL>jar:file:/C:\atfp/images/images.jar!/RHIBIcon.gif</StringsURL>
    </Secondary2DView>
</ViewProperties>
</Entity>
<Entity>
    <EntityProperties>
        <Name>terroristSmallBoat</Name>
        <Identification>Terrorist Small Boat</Identification>

```

```

<Length>10.7</Length>
<Width>1.0</Width>
<Height>1.3</Height>
<MinimumSpeedElement>-6.0</MinimumSpeedElement>
<MaximumSpeedElement>30.0</MaximumSpeedElement>
<EntityPosition>
  <XYZPosition>5905.4 0 5957.6</XYZPosition>
  <GISStringAnchorLatitude>34 00.0N</GISStringAnchorLatitude>
  <GISStringAnchorLongitude>119 00W</GISStringAnchorLongitude>
  <XYPosition>522.0 166.0</XYPosition>
</EntityPosition>
<EntityRateOfTurn>0.1058</EntityRateOfTurn>
<EntityHeading>2.443460952792061</EntityHeading>
</EntityProperties>
<ControlModeProperties>Agent Controlled</ControlModeProperties>
<NetworkProperties>
  <NetworkProtocolPrimary>DIS</NetworkProtocolPrimary>
  <IPAddress>127.0.0.1</IPAddress>
  <DISInformation>
    <EntityIdentification>65</EntityIdentification>
    <SiteIdentification>0</SiteIdentification>
    <ApplicationIdentification>1</ApplicationIdentification>
    <EntityKind>1</EntityKind>
    <EntityCategory>0</EntityCategory>
    <EntityDomain>3</EntityDomain>
  </DISInformation>
</NetworkProperties>
<TacticalProperties>
  <WeaponProperties>
    <ProbabilityOfWeaponSuccess>1.0</ProbabilityOfWeaponSuccess>
    <WeaponName>TNT</WeaponName>
    <WeaponModel>TNT Equivalent</WeaponModel>
    <WeaponAmount>0.3</WeaponAmount>
    <WeaponLethalityRange>20.0</WeaponLethalityRange>
  </WeaponProperties>
  <LineOfSiteDistance>6000.0</LineOfSiteDistance>
  <LineOfSiteProbabilityOfSiting>1.0</LineOfSiteProbabilityOfSiting>
</TacticalProperties>
<ViewProperties>
  <Primary3DView>

```

```

<StringsURL>C:\atfp/resources/org/npsnet/v/views/platforms/surface/TerroristBoatPrototypeNPSNET.wrl</StringsURL>

```

```

        </Primary3DView>
        <Secondary3DView>

<StringsURL>http://web.nps.navy.mil/~brutzman/Savage/Scenarios/TerroristBoatPrototype.wrl</StringsURL>
        </Secondary3DView>
        <Primary2DView>
            <StringsURL>C:\atfp/images/TerroristIcon.gif</StringsURL>
        </Primary2DView>
        <Secondary2DView>
            <StringsURL>jar:file:/C:\atfp/images/images.jar!/TerroristIcon.gif</StringsURL>
        </Secondary2DView>
    </ViewProperties>
</Entity>
</TacticalData>
<StatisticOutput>
    <MOE name="Plan Effectiveness" description="">
        <OverAllScenarios TotalNumberRuns="10.0" RandomNumberSeed="1003.0" DefensePercentEffective="20.0"
Mean="0.2" MeanPercentEffective="20.0" StandardDeviation="0.4216370213557839" Variance="0.1777777777777778"/>
        <ScenarioRun value="1">
            <DefensePercentEffective value="0.0"/>
            <MeanPercentEffective value="0.0"/>
            <StandardDeviation value="0.0"/>
            <Variance value="0.0"/>
        </ScenarioRun>
        <ScenarioRun value="2">
            <DefensePercentEffective value="0.0"/>
            <MeanPercentEffective value="0.0"/>
            <StandardDeviation value="0.0"/>
            <Variance value="0.0"/>
        </ScenarioRun>
        <ScenarioRun value="3">
            <DefensePercentEffective value="0.0"/>
            <MeanPercentEffective value="0.0"/>
            <StandardDeviation value="0.0"/>
            <Variance value="0.0"/>
        </ScenarioRun>
        <ScenarioRun value="4">
            <DefensePercentEffective value="0.0"/>
            <MeanPercentEffective value="0.0"/>
            <StandardDeviation value="0.0"/>
            <Variance value="0.0"/>
        </ScenarioRun>
    </MOE>
</StatisticOutput>

```

```

<ScenarioRun value="5">
  <DefensePercentEffective value="0.0"/>
  <MeanPercentEffective value="0.0"/>
  <StandardDeviation value="0.0"/>
  <Variance value="0.0"/>
</ScenarioRun>
<ScenarioRun value="6">
  <DefensePercentEffective value="0.0"/>
  <MeanPercentEffective value="0.0"/>
  <StandardDeviation value="0.0"/>
  <Variance value="0.0"/>
</ScenarioRun>
<ScenarioRun value="7">
  <DefensePercentEffective value="0.0"/>
  <MeanPercentEffective value="0.0"/>
  <StandardDeviation value="0.0"/>
  <Variance value="0.0"/>
</ScenarioRun>
<ScenarioRun value="8">
  <DefensePercentEffective value="12.5"/>
  <MeanPercentEffective value="12.5"/>
  <StandardDeviation value="0.3535533905932738"/>
  <Variance value="0.125"/>
</ScenarioRun>
<ScenarioRun value="9">
  <DefensePercentEffective value="22.22222222222222"/>
  <MeanPercentEffective value="22.22222222222222"/>
  <StandardDeviation value="0.44095855184409843"/>
  <Variance value="0.19444444444444445"/>
</ScenarioRun>
<ScenarioRun value="10">
  <DefensePercentEffective value="20.0"/>
  <MeanPercentEffective value="20.0"/>
  <StandardDeviation value="0.4216370213557839"/>
  <Variance value="0.17777777777777778"/>
</ScenarioRun>
</MOE>
</StatisticOutput>
</Head>
</ATFPScenarioType>

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [Bourret 2003] Ronald Bourret, January 2003,
<http://www.rpbouret.com/xml/XMLAndDatabases.htm>
- [Carlson 1998] Carlson, Dave; Component Interoperability with XML; appeared
in component Strategies, October 19998; Column: Objects & the Web
- [Cham 2002] Chami.com 2003, <http://www.chami.com/html-kit>
- [Cham 2002] Denis Howe 1998,
<http://burks.brighton.ac.uk/burks/foldoc/subjects/33.htm>
- [DEFENSE 1999] Department of Defense; MIL-STD-2525B, 30 January 1999.
- [Hall 2000] Hall, Marty; Core Servlets and JavaServer Pages, 2000, Sun
Microsystems Press, Upper Saddle River, NJ.
- [Harney 2003] Harney, James. Analyzing Tactical Effectiveness for Anti-
Terrorist Force Protection (AT/FP) Using Using X3D Graphics and
Agent-Based Simulation, Master's Thesis, Naval Postgraduate School.
Monterey, CA, March 2003.
- [Hunter 2001] Hunter, David; Cagle, Kurt; Dix, Chris; Kovack, Roger; Pinnock,
Jonathan, Rafter, Jeff; Beginning XML (2nd Edition), 2001, Wrox Press,
Birmingham, UK.
- [Icarus 2002] Icarus 2002, <http://www.devshed.com/Authors/icarus/>
- [Kay 2001] Kay, Michael, 2001,
<http://www-106.ibm.com/developerworks/xml/library/x-xslt2/>.
- [Laurie 1999] Ben Laurie & Peter Laurie; Apache the definitive guide (2nd
Edition), 1999, O'REILLY & Associates, Sebastopol, California.

[NATO 2002] NATO, “Overview of the Land C2 Information Exchange Data Model (LC2IEDM)”, 18 March 2000.

[Nicklaus 2001] Nicklaus, S. (2001). Scenario Authoring and Visualization for Advanced Graphical Environments. Master’s Thesis, Naval Postgraduate School.

[Valikov 2001] Aleksei Valikov <http://herakles.fzi.de/GuestBook>, Nov 2001.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
For Belvoir Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Don Brutzman
Naval Postgraduate School
Monterey, California
4. Curt Blais
Naval Postgraduate School
Monterey, California
5. Erik Chaum
NAVSEA Undersea Warfare Center
Division Newport
Code 2231, Building 1171-3
Newport, RI
6. Dr. Francisco Loaiza and Dr. Eugene
Simaitis Institute for Defense Analyses
Systems Evaluation Division
Alexandria, VA